

AFIT/GAE/ENY/98M-02

**AN EVALUATION OF FREQUENCY DOMAIN
ENSEMBLE AVERAGING TO IMPROVE
AIRCRAFT STABILITY DERIVATIVE ESTIMATION**

THESIS

Lawrence M. Hoffman, Captain, USAF

AFIT/GAE/ENY/98M-02

Approved for public release; distribution unlimited

19980423 069

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GAE/ENY/98M-02

**AN EVALUATION OF FREQUENCY DOMAIN ENSEMBLE AVERAGING
TO IMPROVE AIRCRAFT STABILITY DERIVATIVE ESTIMATION**

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Lawrence M. Hoffman

Captain, USAF

March 1998

Approved for public release; distribution unlimited

**A LIMITED EVALUATION OF THE HAVE DERIVATIVES PROCESS
TO REDUCE AIRCRAFT STABILITY DERIVATIVE ESTIMATION
ERRORS CAUSED BY PROCESS AND MEASUREMENT NOISE**


Lawrence M. Hoffman, B.S.
Captain, USAF

Degree of Master of Science in Aeronautical Engineering

March 1998

Approved:


Date


Dr. Brad S. Liebst (Chairman)

3/16/98


Dr. Tom Buter

3/5/98


Dr. Sharon A. Heise

3/16/98

Acknowledgments

A project of this magnitude would not have been possible without much help from others. Thank you to my advisor, Dr. Brad Liebst, who originated the idea for this research and provided just the right amount of guidance and encouragement along the way, and to LtCol. Tom Buter for keeping me on track as I wound up the project at the US Air Force Test Pilot School (TPS).

A drawback of the joint Air Force Institute of Technology / TPS program was the geographical separation from my advisor during some of the most critical periods of the research. I was fortunate to have Mr. Chris Nagy, Mr. Tom Twisdale, and Mr. Ralph Smith as surrogate advisors during the flight test portion of this project. Their breadth and depth of flight test experience were greatly appreciated and directly responsible for some of the breakthroughs interpreting the flight test and parameter estimation results.

The flight test program could not have been accomplished without the Have Derivatives team. Maj. Mike Masucci, staff monitor, Capt. Malt Verburg (Royal Netherlands Air Force), project pilot, and Captains Joel Hagan, Guy Eshel (Israeli Air Force), and Hubert L'Ebraly (French Department of Defense), project engineers, comprised a most able international team. Their considerable contributions reflected great credit upon themselves and their countries. I am fortunate indeed to have had the opportunity to work with them.

Finally, I would not have survived this project without the constant support and encouragement of my wife, Jill, and daughter, Noelle. Thank you both for always being there when I needed you!

Lars Hoffman

Table of Contents

	Page
Acknowledgments	iii
List of Figures	vii
List of Tables	ix
List of Acronyms and Symbols	xi
Abstract	xv
 I. Introduction	 1.1
Overview	1.1
General Parameter Estimation Process	1.2
Parameter Estimation Methods	1.4
Research Motivation	1.5
Research Outline	1.7
Research Objectives	1.7
Research Approach	1.8
Summary	1.9
Preview	1.9
 II. Theoretical Development	 2.1
Overview	2.1
The Output-Error Parameter Estimation Method	2.2
Equations of Motion	2.3
Cost Function	2.4
Minimization Algorithm	2.4
Personal Computer Parameter Identification (PCPID)	2.5
Ensemble Averaging	2.6
The Have Derivatives Process	2.8
Average Frequency Response Estimation	2.9
Mirroring	2.11
Inverse Transformation	2.11
Summary	2.12
 III. Simulation Development	 3.1
Overview	3.1
Sign and Units Convention	3.2

Basic Model Development	3.3
Process Noise Model Development	3.7
Measurement Noise Model Development	3.9
Complete Model and Simulation Program	3.11
Summary	3.13
 IV. Simulation Results and Analysis	 4.1
Overview	4.1
Simulation Results From PCPID Alone	4.2
Process Noise Only Simulation Results	4.3
Measurement Noise Only Simulation Results	4.6
Combined Process and Measurement Noise Simulation Results	4.9
No Noise Simulation Results Using Different Input Functions	4.11
Simulation Results From the Have Derivatives Process With PCPID	4.14
Overlap Percent Simulation Results	4.15
Section Length Simulation Results	4.17
Maneuver Length Simulation Results	4.18
Summary	4.20
 V. Flight Test Development	 5.1
Overview	5.1
Introduction	5.1
Test Aircraft Description	5.1
Flight Test Objectives	5.2
Flight Test Procedures	5.3
Test Aircraft Configuration	5.3
Flight Test Inputs	5.5
Test Point Flight Conditions	5.10
Data Reduction Procedures	5.11
Summary	5.14
 VI. Flight Test Results and Analysis	 6.1
Overview	6.1
Calm Air Results From PCPID Alone (Validation)	6.1
Calm Air Results From PCPID Alone (Flight Test Basis Data Set)	6.3
Calm Air Results From the Have Derivatives Process With PCPID	6.9
Turbulent Air Results From PCPID Alone	6.21
Turbulent Air Results From the Have Derivatives Process With PCPID	6.26
Summary	6.33
 VII. Conclusions and Recommendation	 7.1
Overview	7.1
Simulation Conclusions	7.1

Flight Test Conclusions	7.2
Recommendation for Further Research	7.3
Appendix A: Have Derivatives Simulation Program	A.1
Appendix B: Simulation Results	B.1
Appendix C: Data Reduction Programs	C.1
Have Derivatives Data Reduction Program	C.1
Mass Characteristics Program	C.21
Appendix D: Calm Air Stability Derivative Estimation Results	D.1
Appendix E: Turbulent Air Stability Derivative Estimation Results	E.1
Bibliography	BIB.1
Vita	VIT.1

List of Figures

Figure	Page
1.1 General Aircraft Parameter Estimation Process	1.3
2.1 Output-Error Parameter Estimation Method	2.2
3.1 Simulation and Flight Test Sign Convention	3.2
3.2 Measurement Noise Model	3.10
4.1 Doublet Responses, Ideal and Measured with 3 ft/sec Turbulence Intensity . . .	4.4
4.2 Doublet Responses, Ideal and Measured with 9 ft/sec Turbulence Intensity . . .	4.4
4.3 Root-Sum-of-Squares Magnitude versus Root-Mean-Square Turbulence Intensity	4.5
4.4 Doublet Responses, Ideal and Measured with 10% Measurement Noise	4.7
4.5 Doublet Responses, Ideal and Measured with 30% Measurement Noise	4.7
4.6 Root-Sum-of-Squares Magnitude versus Measurement Noise Level	4.8
4.7 Doublet Responses, Ideal and Measured with Calm Conditions	4.10
4.8 Doublet Responses, Ideal and Measured with Turbulent Conditions	4.10
4.9 Root-Sum-of-Squares Magnitude versus Combined Conditions	4.11
4.10 Elevator Deflection versus Time for Various Input Functions	4.12
4.11 Root-Sum-of-Squares Magnitude versus Input Function	4.13
4.12 Root-Sum-of-Squares Magnitude versus Percent Overlap	4.16
4.13 Root-Sum-of-Squares Magnitude versus Section Length	4.17
4.14 Root-Sum-of-Squares Magnitude versus Maneuver Length	4.19
4.15 Turbulent Conditions, Impulse Response Without Have Derivatives Processing .	4.21
4.16 Turbulent Conditions, Discrete Pulse Response From Have Derivatives Process .	4.21

6.1	Example F-16B Calm Air Stability Derivative Estimates and Cramer-Rao Bounds From PCPID Alone	6.6
6.2	Example F-16B Calm Air Time History Matches From PCPID Alone	6.7
6.3	F-16B Simulated Ideal Impulse Responses and Discrete Pulse Responses From Simulated Data	6.10
6.4	Matches of PCPID Computed Time Histories to F-16B Discrete Pulse Responses From Simulated Data	6.11
6.5	F-16B Calm Air Discrete Pulse Responses From the Have Derivatives Process .	6.13
6.6	Matches of PCPID Computed Time Histories to F-16B Calm Air Discrete Pulse Responses From the Have Derivatives Process	6.14
6.7	Matches of PCPID Computed Time Histories to Filtered F-16B Calm Air Discrete Pulse Responses From the Have Derivatives Process	6.18
6.8	Example F-16B Turbulent Air Stability Derivative Estimates and Cramer-Rao Bounds From PCPID Alone	6.23
6.9	Example F-16B Turbulent Air Time History Matches From PCPID Alone . . .	6.24
6.10	F-16B Noise Free and Noisy Simulated Angle of Attack Responses	6.26
6.11	F-16B Turbulent Air Discrete Pulse Responses From the Have Derivatives Process	6.28
6.12	Matches of PCPID Computed Time Histories to Filtered F-16B Turbulent Air Discrete Pulse Responses From the Have Derivatives Process	6.29

List of Tables

Table	Page
3.1 AT-37B and F-16B Approximate Measurement Uncertainties	3.9
4.1. Stability Derivative Weights	4.2
5.1. Project HAVE DERIVATIVES Test Aircraft Configuration	5.4
5.2. Project HAVE DERIVATIVES Turbulence Level Definitions	5.5
5.3. Project HAVE DERIVATIVES Trim Shot Data Bands and Tolerances	5.6
5.4. Project HAVE DERIVATIVES Parameters Recorded During Flight	5.9
6.1. Validation of Data Reduction Processes and F-16B Test Aircraft Instrumentation	6.2
6.2. F-16B Calm Air Mean Estimate Values and Confidence Measures From PCPID Alone	6.4
6.3. F-16B Calm Air Standard Deviations and Mean Cramer-Rao Bounds From PCPID Alone	6.5
6.4. F-16B Response Weighting and Calm Air Mean Cost Function Values From PCPID Alone	6.8
6.5. F-16B Noise Free Simulation, Actual and Estimated Derivative Values and Cramer-Rao Bounds From Have Derivatives Process With PCPID	6.12
6.6. F-16B Calm Air Results From the Have Derivatives Process With PCPID and Associated Flight Test Basis Data Set Results	6.15
6.7. F-16B Filtered Calm Air Results From the Have Derivatives Process With PCPID and Associated Flight Test Basis Data Set Results	6.19
6.8. F-16B Turbulent Air Mean Stability Derivative Estimate Values and Confidence Intervals From PCPID Alone	6.21
6.9. F-16B Turbulent Air Stability Derivative Standard Deviations and Mean Cramer-Rao Bounds From PCPID Alone	6.22
6.10. F-16B Mean Cost Function Values From PCPID Alone	6.25

6.11.	F-16B Filtered Results From the Have Derivatives Process With PCPID and Associated Flight Test Basis Data Set Results	6.30
6.12.	F-16B Filtered Turbulent Air Results From the Have Derivatives Process With PCPID, Turbulent Air Doublet Results From PCPID Alone, and Associated Flight Test Basis Data Set Results	6.31
B.1.	Process Noise Only Simulation Results	B.1
B.2.	Measurement Noise Only Simulation Results	B.2
B.3.	Combined Process and Measurement Noise Simulation Results	B.2
B.4.	No Noise Simulation Results Using Different Input Functions	B.3
B.5.	Percent Overlap Simulation Results	B.3
B.6.	Ideal Conditions Section Length Simulation Results	B.4
B.7.	Calm Conditions Section Length Simulation Results	B.4
B.8.	Turbulent Conditions Section Length Simulation Results	B.5
B.9.	Maneuver Length Simulation Results	B.5
D.1.	Turbulence Level Definitions	D.1
D.2.	Calm Air Stability Derivative Estimate Values and Cramer-Rao Bounds	D.3
D.3.	Calm Air Time History Match Cost Function Values	D.4
E.1.	Turbulence Level Definitions	E.1
E.2.	Turbulent Air Stability Derivative Estimate Values and Cramer-Rao Bounds . .	E.3
E.3.	Turbulent Air Time History Match Cost Function Values	E.4

List of Acronyms and Symbols

Symbol	Definition	Units
$\hat{H}(\omega)$	average complex frequency response function estimate	
$\dot{\alpha}$	derivative of angle of attack with respect to time	(deg / sec)
\dot{q}	derivative of pitch rate with respect to time	(deg / sec)
\bar{q}	dynamic pressure	(lb _f / ft ²)
\tilde{z}	response computed by integrating the equations of motion	
$\hat{P}_{xx}(\omega)$	average auto-spectral density estimate of the input	
$\hat{P}_{xy}(\omega)$	average cross-spectral density estimate between the input and response	
Δg	Δ Normal Acceleration	
AFBTC	Air Force Flight Test Center	
AOA	angle of attack	
ASCII	American Standard Code Information Interchange	
ATIS	airborne test instrumentation system	
AUTO	automatic	
b	wing span	(ft)
BUC	back up control	
c	aircraft reference chord	(ft)
CADC	central air data computer	
C_{mq}	coefficient of pitching moment due to pitch rate	(per rad)
$C_{m\alpha}$	coefficient of pitching moment due to angle of attack	(per deg)

$C_{m\delta e}$	coefficient of pitching moment due to elevator control	(per deg)
$C_{N\alpha}$	coefficient of normal force due to angle of attack	(per deg)
$C_{N\delta e}$	coefficient of normal force due to elevator control	(per deg)
DAS	data acquisition system	
deg	degree(s)	
EEC	electronic engine control	
FFT	fast Fourier transform	
fps	feet per second	
ft	foot / feet	
FTAS	Flight Test Analysis Software	
g	acceleration due to gravity (32.174)	(ft / sec ²)
G_{mn}	high-pass Butterworth filter	
HUD	head up display	
in Hg	inches of Mercury	
INS	inertial navigation system	
IFFT	inverse fast Fourier transform	
I_{xy}	aircraft product of inertia about the $X_b Y_b$ axis	(slugs / ft ²)
I_{yz}	aircraft product of inertia about the $Y_b Z_b$ axis	(slugs / ft ²)
I_{yy}	aircraft moment of inertia about the Y_b axis	(slugs / ft ²)
K_i	measurement uncertainty scale factor ($i = \alpha, q$ or δe)	
KTAS	knots true airspeed	
L	turbulence scale length	(ft)
lb	pound(s)	

LVDT	linear variable displacement transducer	
m	aircraft mass	(slugs)
M	Mach number	
NASA	National Aeronautics and Space Administration	
n_i	number of time history points ($i = 1, 2, \dots$)	
NORM	normal	
n_z	number of response variables ($z = 1, 2, \dots$)	
PA	pressure altitude	
PCPID	Personal Computer Parameter Identification	
pct MAC	percent of mean aerodynamic chord	
PEST	parameter estimation	
PNEU	pneumatic	
q	pitch rate	(deg / sec)
R	conversion factor (~ 57.3)	(deg)
rms	root-mean-square	
rss	root-sum-of-squares	
S	wing area	(ft ²)
T	denotes transpose	
t_i	time ($i = 1, 2, \dots$)	(sec)
TPS	Test Pilot School	
USAF	United States Air Force	
ξ	vector of unknown parameters	
V_t	aircraft true velocity	(ft / sec)

W	response weighting matrix	
x_{cg}	center of gravity distance from reference along X_b (positive aft)	(ft)
x_{nz}	normal accelerometer distance from reference along X_b (positive aft)	(ft)
x_a	angle of attack sensor distance from reference along X_b (positive aft)	(ft)
z	measured aircraft response	
α	angle of attack	(deg)
α_0	angle of attack at time $t = 0$	(deg)
δe	elevator or horizontal stabilator deflection	(deg)
η	zero mean, Gaussian distributed, band-limited white noise	
v_i	scaled measurement noise ($i = \alpha, q$ or δe)	
σ	root-mean-square (rms) turbulence intensity	(ft / sec)
ω	angular frequency	(rad / sec)

Abstract

This research evaluated a process to improve aircraft stability derivative estimation results. The Have Derivatives process used overlap ensemble averaging in the frequency domain to minimize noise on the original time domain signals. The process estimated average complex frequency response functions that were then transformed back into the time domain as a set of discrete pulse responses with far less noise than the original signals. These 'clean' signals were used in a parameter estimation program to estimate better stability derivatives than were estimated with the original noisy signals.

Both simulation and flight test data were used to study the effects of various noise levels on stability derivative estimation results and to evaluate the Have Derivatives process to improve those results. The simulations demonstrated dramatic improvement using the Have Derivatives process. The flight test results were not as conclusive.

The ensemble averaging step of the Have Derivatives process was not effective enough at reducing noise on the flight test data due to non-uniform frequency content of the flight test input. The overall recommendation was to further evaluate the Have Derivatives process using a broadband flight test input, similar to the input that worked well in simulation.

AN EVALUATION OF FREQUENCY DOMAIN ENSEMBLE AVERAGING TO IMPROVE AIRCRAFT STABILITY DERIVATIVE ESTIMATION

1. Introduction

1.1 Overview

Process noise and measurement noise are always present to varying degrees on aircraft input and response signals measured by onboard sensors and recorded by a data acquisition system. The noise makes it difficult for parameter estimation routines to correlate aircraft responses with the input(s) that forced those responses. This can reduce the accuracy and precision of the aircraft stability derivatives estimated by those routines.

The body of this introductory chapter opens with a discussion of the general parameter estimation process. This discussion is followed by a presentation of three methods that have been developed to accomplish aircraft parameter estimation as a subset of the general process. The primary difference between these methods is how they account for process and measurement noise. Of the three discussed, the output-error method is the most widely used and accepted.

Next the motivation for this research effort is presented, an opportunity to improve stability derivative estimation results using the Have Derivatives process with the output-error method.

Finally, the objectives of this research effort and the approach taken to satisfy those objectives are outlined.

1.2 General Parameter Estimation Process

Parameter identification is a subset of a larger field of study known as system identification. System identification techniques have been used widely in attempts to answer the inverse question, 'Given the answer, what is the question?' (Hamel and Jategaonkar, 1996:9). Through time, humanity has attempted to understand and master the environment through observing and characterizing the systems that compose that environment. The system identification problem views physical systems as 'black boxes' with no specified form. This nonparametric approach has many useful applications but its shortcoming is that it provides the investigator no information about the internal system structure (Maine and Iliff, 1985:3). Spectral analysis is one nonparametric system identification method that was used in this research to estimate system frequency responses (Ljung, 1995:1-3). This application will be explored more in Chapter 2, Theoretical Development.

Rather than viewing physical systems as black boxes, the parameter identification approach assumes a general form of the physical system based on knowledge of the system (Maine and Iliff, 1985:4 to 5). In the realm of aircraft parameter identification this form becomes the aircraft equations of motion. The equations of motion describe the bare airframe response to control surface deflections, changes in thrust and atmospheric disturbances. The equations are composed of scalar terms, based on the aircraft physical properties and flight condition, and stability derivatives, variables that model the aircraft aerodynamics. The derivatives describe the change in aerodynamic forces and moments with changes in aircraft attitude, angular rates, or control surface deflections. Given the scalar values, current aircraft states, and control surface deflections, parameter identification algorithms attempt to determine the stability derivative values.

However, because of modeling errors, measurement errors, and noise, the algorithms can only estimate the stability derivative values, not identify them exactly. For that reason, the algorithms are more appropriately referred to as parameter estimation methods (Ilyf, 1987:2).

A general parameter estimation process is presented in Figure 1.1.

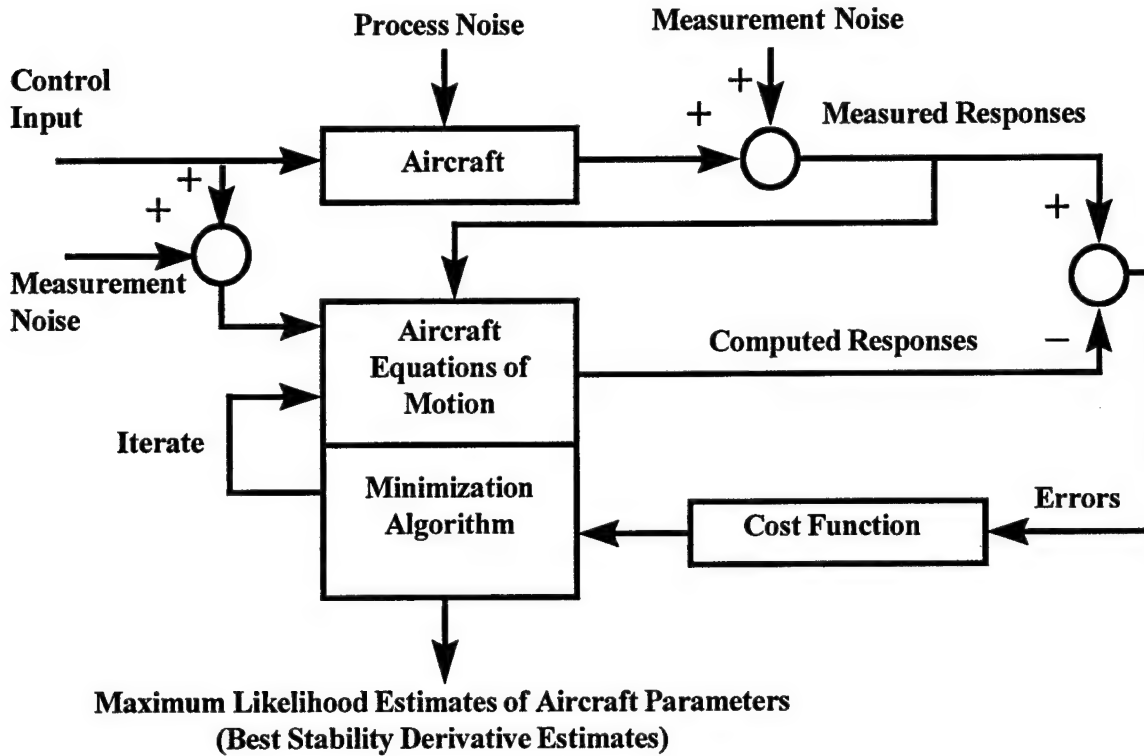


Figure 1.1. General Aircraft Parameter Estimation Process (Ilyf, 1987:18)

The estimation process shown in Figure 1.1 is straightforward. Responses are computed from the equations of motion using the same control input made to the aircraft. These computed responses are compared with the measured aircraft responses. The difference between the measured and computed responses is quantified by the cost function. If the cost function does not satisfy convergence criteria, the process continues with the minimization algorithm. The

minimization algorithm searches for parameter values that minimize the cost function and updates the equations of motion with the new parameter values. New responses are computed from the updated equations of motion and the original control input. The process iterates, updating the parameters each time, until the cost function convergence criteria is met (Iliff, 1987:6). The parameter values upon convergence are the maximum likelihood, or best, estimates of the aircraft parameters, including the stability derivatives (Iliff, 1987:2).

1.3 Parameter Estimation Methods

The most widely used parameter estimation methods can be classified into three broad approaches which are variations of the general process shown in Figure 1.1. The choice of which method to use depends upon model formulation and assumptions made regarding the measurement and process noise (Hamel and Jategaonkar, 1996:13). The three methods are known as filter-error, equation-error and output-error.

The filter-error method is the most general approach and takes the form shown in Figure 1.1. The filter-error method accounts for both process and measurement noise but is not widely used due to the computational complexity of implementing the method. To simplify computations when using the filter-error method, non-linear systems with both process and measurement noise are typically linearized. This approach makes the computations tractable but gives less accurate results for non-linear systems (Maine and Iliff, 1985:95).

One subset of the filter-error method is the equation-error approach which accounts for process noise but ignores measurement noise. The equation-error method uses linear regression to estimate stability derivatives. The linear regression based equation-error approach greatly simplifies the problem but its weakness is that relatively small amounts of measurement noise

can cause significant bias errors in the estimates (Maine and Iliff, 1985:105; Hamel and Jategaonkar, 1996:13).

The other subset of the filter-error method is the output-error method. In contrast to the equation-error method, the output-error method accounts for measurement noise on the aircraft response signals but does not account for process noise on the responses or measurement noise on the control input signal. The output-error method greatly simplifies computations while being applicable to both linear and nonlinear systems (Klein, 1989:53). The nonlinear capability was an improvement over earlier maximum likelihood implementations (Murray and Maine, 1987:1).

The output-error method has been the most widely accepted and used of the three methods since its introduction in the 1970's. The output-error method is also the algorithm coded in the parameter estimation software currently used throughout the National Aeronautics and Space Administration (NASA) Dryden Flight Research Center and the Air Force Flight Test Center (AFFTC) (Flying Qualities Branch, 1992:24-8).

1.4 Research Motivation

The output-error method will be discussed in more detail in Chapter 2 but its major shortcoming has already been addressed. The output-error method does not account for process noise (Maine and Iliff, 1985:88; Maine and Iliff, 1986:2). Process noise is the random excitation of the system from unmeasured sources. The standard example of process noise for aircraft parameter estimation applications is atmospheric disturbances, or turbulence (Iliff, 1987:4). Turbulence affects the aircraft responses directly and randomly. Because the output-error method does not account for process noise, the algorithm is unable to correlate the random variations of the aircraft responses, due to turbulence, with the control surface deflections, which are not affected the same by the turbulence. Consequently, reliable stability derivatives cannot

be estimated from data gathered in turbulence. Even light turbulence has been shown to have a noticeably deleterious effect on derivative estimate accuracy and uncertainty level (Flying Qualities Branch, 1992:24-30 to 24-31). The current way of dealing with this deficiency in the output-error method is to accomplish parameter estimation flights in calm air.

This shortcoming of the output-error method was the motivation for this research. Previous research showed the effectiveness of ensemble averaging in the frequency domain to reduce random signal variance (Welch, 1967; Porat, 1994:112 to 114; Kay, 1988:63 to 94). The goal of this research was to evaluate the effectiveness of ensemble averaging to reduce signal variance caused by both measurement and process noise. It was expected that reducing signal variance would improve stability derivative estimation accuracy and precision using the output-error parameter estimation method.

The output-error algorithm is coded in the parameter estimation software, Personal Computer Parameter Identification (PCPID), currently used by the USAF Test Pilot School (TPS) (Quartic Engineering Inc., 1997:1). Consequently, a new method to improve output-error results would provide a tangible benefit to the Test Pilot School by upgrading existing parameter estimation capabilities. This secondary research goal provided additional motivation.

The challenge was that ensemble averaging is accomplished in the frequency domain while PCPID required aircraft input and response data in the time domain to perform parameter estimation using the output-error algorithm. The Have Derivatives process was developed to transform time domain data into the frequency domain, where ensemble averaging could be accomplished, followed by transformation back into the time domain for PCPID parameter estimation. The details of the process will be covered in Chapter 2.

1.5 Research Outline

1.5.1 Research Objectives. An overall research objective and several specific objectives were defined to reach the goals expressed in the previous subsection. The overall objective was to perform a limited evaluation of the Have Derivatives process to reduce aircraft stability derivative estimation errors caused by process and measurement noise.

The overall objective was further defined by the following specific research objectives:

1. Establish a simulation baseline set of estimation results from simulated data with no process or measurement noise, using a doublet input and PCPID alone.
2. Demonstrate the degrading effects of process and measurement noise in simulation using a doublet input and PCPID alone.
3. Compare stability derivative estimation results from simulated data with no process or measurement noise, using different input functions, to the simulation baseline results.
4. Determine the optimal combination of section overlap, section length, and maneuver length, from simulated data with various noise levels, using the Have Derivatives process with PCPID.
5. Establish a flight test basis data set of stability derivative estimation results from calm air flight test data using a doublet input and PCPID alone.
6. Compare stability derivative estimation results from calm air flight test data, using the Have Derivatives process with PCPID, to the flight test basis data set results.
7. Compare stability derivative estimation results from turbulent air flight test data, using PCPID alone, and using the Have Derivatives process with PCPID, to the flight test basis data set results and to each other.

1.5.2 Research Approach. The following approach was developed to satisfy each of the research objectives:

1. Develop a basic aircraft model for computer simulation from the longitudinal short period approximation equations of motion.
2. Augment the basic aircraft model with the capability to simulate variable levels of process noise, in the form of turbulence, using the Dryden vertical disturbance form.
3. Develop a measurement noise model and include the capability to add variable levels of measurement noise to the simulation input and response signals.
4. Accomplish a baseline simulation with no noise, using a doublet input.
5. Accomplish simulations with increasing levels of turbulence and measurement noise, using a doublet input.
6. Accomplish simulations with no noise, using different input functions.
7. Accomplish simulations with various noise levels, using the Have Derivatives process, with different combinations of section overlap, section length, and maneuver length.
8. Process simulation data with PCPID to produce stability derivative estimation results.
9. Compare results from subsequent simulation runs to the simulation baseline results.
10. Process calm air flight test data using PCPID alone and create a basis data set of results.
11. Process calm air flight test data using the Have Derivatives process with PCPID and compare the results to the flight test basis data set.
12. Process turbulent air flight test data using PCPID alone and compare the results to the flight test basis data set.
13. Process turbulent air flight test data using the Have Derivatives process with PCPID and compare the results both to the flight test basis data set and to the turbulent air results found using PCPID alone.

1.6 Summary

This chapter first introduced the general parameter estimation process and discussed three methods which are variations of that general form.

The motivation for this research was presented next as an opportunity to improve stability derivative estimation results using the Have Derivatives process with PCPID, a parameter estimation software package that uses the output-error parameter estimation method.

Finally, a research outline was given, including objectives and an approach to meet those objectives.

1.7 Preview

Chapter 2 covers the theoretical background necessary to understanding the information presented in subsequent chapters.

The program used to evaluate the Have Derivatives process in simulation is developed in Chapter 3, Simulation Development.

Simulation results from the program developed in Chapter 3 are discussed and analyzed in the fourth chapter. These results include both simulation products and the stability derivative estimation results from PCPID alone and from the Have Derivatives process with PCPID.

Chapter 5, Flight Test Development, outlines a flight test program to evaluate the Have Derivatives process using flight test data.

Stability derivative estimation results, from the flight test program outlined in Chapter 5, are presented and analyzed in Chapter 6.

The final chapter summarizes conclusions from the first six chapters and offers recommendations for continued research.

2. Theoretical Development

2.1 Overview

This chapter introduces the theoretical background necessary to understanding the information presented in subsequent chapters. A build-up approach is taken by starting with the component parts and developing to the complete processes that were used in this research.

The output-error parameter estimation method, introduced in Chapter 1, will be examined in greater detail in the first main subsection. The main components of the output-error method are the aircraft equations of motion, the cost function, and the minimization algorithm. Each of these components are described as implemented by the parameter estimation program at the heart of the Personal Computer Parameter Identification (PCPID) software used during this research.

The next subsection presents an overview of PCPID and its five subroutines designed to aid the user in accomplishing parameter estimation.

The theory behind ensemble averaging is explained in subsection 2.4 to lay groundwork for the subsequent Have Derivatives process development in which the ensemble averaging step plays a central role.

The final main subsection presents the complete Have Derivatives process. The process transforms input and response time histories into the frequency domain to estimate average complex frequency response functions using an ensemble averaging technique. The process then transforms the average frequency responses back into the time domain as discrete pulse responses. The final step of the process is to create a simulated ideal impulse input function to be saved with the discrete pulse responses for PCPID processing.

2.2 The Output-Error Parameter Estimation Method

The output-error form of the general parameter estimation process is presented in Figure 2.1.

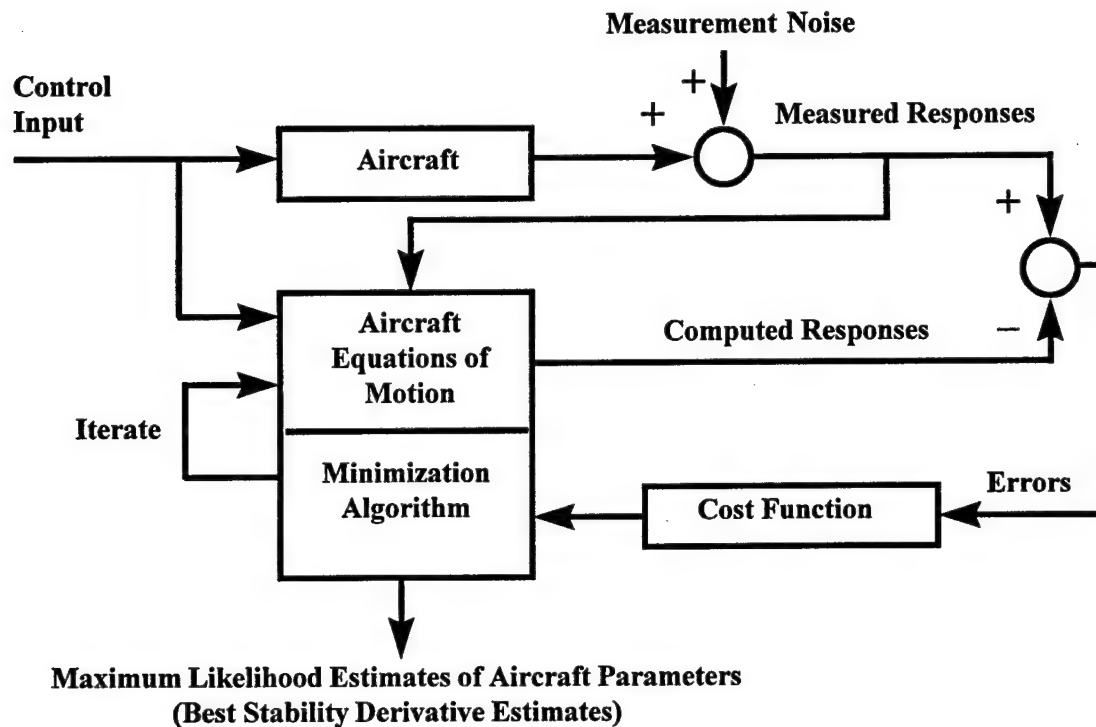


Figure 2.1. Output-Error Parameter Estimation Method

The only differences in Figure 2.1 from the general parameter estimation process, presented in Figure 1.1 of Chapter 1, are the absence of a process noise input to the aircraft block, and the absence of additive measurement noise on the control input signal path leading to the equations of motion block. These changes reflect the fact that the output-error method accounts only for the measurement noise added to the aircraft response signals.

Blocks in Figure 2.1 illustrate the main components of the output-error method; the aircraft equations of motion, the cost function, and the minimization algorithm. The minimization

algorithm can be further subdivided into the minimization method, gradient computation method and convergence bound. These are user selected options that define the iterative execution of the output-error method.

The operation of the output-error method is straightforward. The program computes aircraft responses from the aircraft equations of motion using the aircraft control input. The computed responses are compared to the actual aircraft responses using a quadratic cost function. If the percentage of cost function change is too large, the program changes the vector of unknown parameters in the equations of motion according to a minimization algorithm. The process repeats in an iterative fashion until the agreement between the computed and measured responses satisfies convergence criteria.

Following are descriptions of the main output-error components as implemented by the parameter estimation program in the PCPID software used during this research.

2.2.1 Equations of Motion. The most general aircraft model solved by the program are a vector set of time-varying, finite-dimensional, ordinary differential equations of motion. The equations are composed of a continuous-time state equation and a discrete-time response equation (Murray and Maine, 1987:3). Although the program has the capability to model an aircraft with the full six-degree-of-freedom nonlinear equations of motion, a subset is usually selected by the user (Murray and Maine, 1987:18). A longitudinal two-degree-of-freedom subset of state equations was selected for the iterative process in the current study.

Either an Euler or a fourth-order Runge-Kutta algorithm may be selected by the user to integrate the equations of motion. The Euler algorithm is faster but the Runge-Kutta algorithm is more accurate and has a larger region of stability (Murray and Maine, 1987:17). The Runge-Kutta algorithm was chosen for this research.

2.2.2 Cost Function. The cost function computes a scalar that quantifies the level of agreement between the computed responses and the aircraft's measured responses (Murray and Maine, 1987:2).

The cost function is given in Equation (2.1) as:

$$J(\xi) = \frac{1}{2 n_z n_i} \sum_{i=1}^{n_i} [z(t_i) - \tilde{z}(t_i)]^T W [z(t_i) - \tilde{z}(t_i)] \quad (2.1)$$

where

- ξ = vector of unknown parameters
- n_z = number of response variables
- n_i = number of time history points
- t_i = time (sec)
- z = measured aircraft response
- \tilde{z} = response computed by integrating the equations of motion
- T = denotes transpose
- W = response weighting matrix

2.2.3 Minimization Algorithm. The parameters are varied to minimize the cost function during each iteration of a parameter estimation run according to user selected options. The minimization method, gradient computation method, and gradient computation step size dictate how the parameters are varied in the equations of motion. There are six minimization methods available in PCPID's parameter estimation program. Levenberg-Marquart is the most robust of the six and was selected for this research. A single-sided gradient computation method with a step size of 0.00001 are program defaults and were used for all PCPID runs (Quartic Engineering Incorporated, 1997:C-4 to C-5).

The convergence bound defines when the iterative process can terminate. If the percentage change in the cost function value drops below the user specified bound, convergence is declared, and the iterative process terminates (Murray and Maine, 1987:16). The default value of 0.01 percent was used in this research.

2.3 Personal Computer Parameter Identification (PCPID)

The PCPID software is composed of five programs that aid the user in accomplishing parameter estimation post-flight. The heart of the software is its parameter estimation program which implements the output-error parameter estimation method as previously described. The four other programs in the software package are designed to prepare data for input to the parameter estimation program and to produce output using the results from the parameter estimation run. Each of the five programs are now briefly described (Quartic Engineering Incorporated, 1997:1 to 4).

1. The **PCPID Control Shell** provides a centralized program for control of the parameter estimation process.
2. Estimation runs are initialized using the **Program Status File Manager**. Here parameter values are initialized, constant data is entered and user options are selected.
3. The **Time History Editor** displays aircraft input and response time histories for input into the parameter estimation program. Within the Time History Editor, the user is able to select which section of the time histories are to be used for parameter estimation.
4. The **Parameter Estimation** program at the heart of PCPID implements the output-error method to accomplish parameter estimation using input from the Program Status File Manager and Time History Editor.

5. Results from the Parameter Estimation program are displayed by the **Output Generator**.

Stability derivative estimates are presented in text and plots for analysis. Measured and computed time histories are displayed on the same axes to evaluate the final matches.

The Output Generator can save or print the results.

2.4 Ensemble Averaging

While high frequency measurement noise can usually be minimized with low-pass filters, process noise is typically a low frequency phenomenon that cannot be easily separated from the system responses which are usually in the same low frequency range. This property of process noise prevents the use of simple filters to minimize it.

One property that is common to both process and measurement noise, however, is they are random phenomena that manifest themselves as random variations on the measured signals. Referring to Figure 1.1 of Chapter 1, notice that process noise affects only the measured aircraft responses through an extraneous input to the aircraft. Because the control input contains none of the process noise variations, there is no correlation between the process noise caused variations on the aircraft response signals and the control input. Also note that the measurement noise caused random variations on the input and response signals are different for each signal and are assumed to be uncorrelated. In fact, the output-error method, presented in Figure 2.1, shows the measurement noise adding only to the response signals because the method assumes that the input is known exactly (Maine and Iliff, 1985:88). Because measurement noise variations are always present on the input signal, even with the best instrumentation system, this assumption is another source of stability derivative estimation errors. These observations are motivation for a

technique known as ensemble averaging which seeks to minimize the random variations caused by process and measurement noise while preserving the true system dynamics.

The basic mechanics of ensemble averaging are straightforward. Given the input and response time histories of a system, a set, or ensemble, of equal length time history sections is created by dividing the full time histories into equal parts, typically using a rectangular window (Flying Qualities Branch, 1992:25-59). Next, the estimated auto-spectral density is calculated for each input section by squaring the magnitude of the Fourier transform of the section and scaling the result (Porat, 1994:106). The estimated cross-spectral density between each input and response section pair is computed by scaling the product of the Fourier transformed input section and the complex conjugate of the Fourier transformed output section (Krauss and others, 1994:2-67). The scale factor in both cases is equal to the inverse of the section length and is included to ensure that the estimated spectral densities, or periodograms, are asymptotically unbiased estimates of the true spectral densities (Porat, 1994:106 to 107; Kay, 1988:65 to 66). The respective spectral densities are summed at each frequency and divided by the number of sections in the ensemble. The result is ensemble averaged auto-spectral density and cross-spectral density function estimates (Flying Qualities Branch, 1992:25-57).

There are several variations of the basic ensemble averaging technique. Welch's method of modified periodogram averaging was determined to be the most effective and easily implemented (Porat, 1994:112 to 114). Welch's method incorporates several features that address problems with the basic ensemble averaging technique. One of the problems, created by sectioning a time history with a rectangular window, is the resulting discontinuities at the edges of each section. These discontinuities translate into spectral leakage in the frequency domain (Rameriz, 1985:102 to 109). One way to reduce the edge discontinuities and, consequently, the amount of leakage, is to use a non-rectangular window that tapers smoothly to zero at the edges

(Rameriz, 1985:138-145). The modified periodogram is created by applying a non-rectangular window to the time history sections before computing the periodograms as before (Welch, 1967:70 to 71). The modified periodogram is scaled by the norm of the data window to ensure that the estimate is asymptotically unbiased (Krauss and others, 1994:1-64).

The other main feature of Welch's method is the use of overlapping sections from the original long input and response time histories. This produces more sections to be averaged, resulting in less variance on the average spectral densities (Welch, 1967:71 to 72). A common choice for the amount of overlap is 50 percent of the section length (Porat, 1994:112). There is a practical limit to the amount of overlap that can be used. The ensemble averaging technique assumes that each of the sections in the ensemble is independent. As the amount of overlap increases so does the dependence between sections, violating the assumption. The adopted practice at the Air Force Flight Test Center has been to restrict the amount of overlap to no more than 66 percent (Flying Qualities Branch, 1992:25-60).

2.5 The Have Derivatives Process

As mentioned in Chapter 1, the motivation for this research was the inability of the output-error method to address process noise. The challenge was to develop an approach to minimize process noise before importing the signals into PCPID. The approach, if successful, could augment PCPID to improve parameter estimation results. The approach developed in this research became known as the Have Derivatives process.

The Have Derivatives process can be broken into three interrelated steps. First, input and response time histories are measured and ensemble averaged using the Welch method described earlier. The average complex frequency response function for each input and response

combination is estimated from the average spectral density estimates. The second step of the process is to create full average complex frequency response function estimates by folding and flipping the real and imaginary components of the functions about the Nyquist frequency. The final Have Derivatives process step is to inverse fast Fourier transform (IFFT) the full average complex frequency response function estimates back into the time domain as discrete pulse responses. The Have Derivatives process then creates a simulated ideal impulse input time history that correlates with the discrete pulse responses. The discrete pulse responses and simulated ideal impulse input are saved together for PCPID processing. Each of the three Have Derivatives process steps are now described in detail.

2.5.1 Average Frequency Response Estimation. The first step of the Have Derivatives process is to estimate the average complex frequency response function for each aircraft response to input. Using ensemble averaging, the average complex frequency response functions can be estimated by Equation (2.2) (Krauss and others, 1994:1-72).

$$\hat{H}(\omega) = \frac{\hat{P}_{xy}(\omega)}{\hat{P}_{xx}(\omega)} \quad (2.2)$$

where

- $\hat{H}(\omega)$ = average complex frequency response function estimate
- $\hat{P}_{xy}(\omega)$ = average cross-spectral density estimate between the input and response
- $\hat{P}_{xx}(\omega)$ = average auto-spectral density estimate of the input
- ω = angular frequency (rad / sec)

The routine, TFE.M, in MATLAB's® Signal Processing Toolbox® employs Welch's method to compute the average complex frequency response estimate (Krauss and others, 1994:1-72).

The routine assumes that the system is linear and time invariant. Given the measured input and response signals, the routine uses a four-step algorithm (Krauss and others, 1994:2-218 to 2-221).

1. It detrends sections and then multiplies them by the specified window function.
2. It fast Fourier transforms (FFT) each detrended and windowed section.
3. It estimates the average auto-spectral density of the input and the average cross-spectral densities of each input and response combination, as described earlier.
4. It estimates the average complex frequency response function for each input and response pair as shown in Equation (2.2).

The user dictates the specific operation of TFE through several optional settings. First, the user can select from several detrending options. Mean detrending removes the mean from prewindowed sections and was selected during this research to improve the amplitude resolution of the transformed waveforms (Rameriz, 1985:126 to 127).

Several basic and generalized cosine window functions are available to choose from. The user can select both the window type and length. The Hanning window has been used for many years at the Air Force Flight Test Center and provided a good balance between main lobe width and side lobe height (Flying Qualities Branch, 1992:25-62; Rameriz, 1985:139 to 143). A Hanning window with the same length as the FFT was selected for this research.

The user determines section length by specifying the length of the fast Fourier transform. Fastest execution is achieved with a power of 2 FFT length.

The user can also select the number of samples that each section overlaps. The number of sections that will be averaged is then determined by Equation (2.3).

$$\text{Number of Sections} = \frac{\text{Total Signal Length} - \text{Overlap Length}}{\text{Section Length} - \text{Overlap Length}} \quad (2.3)$$

2.5.2 Mirroring. Fourier transform properties dictate that to achieve a real function in the time domain the complex frequency response function must be composed of an even real part and an odd imaginary part (Brigham, 1988:47). Because the average complex frequency response function estimates calculated in the first step do not satisfy these properties, they must be modified before they are inverse transformed back into the time domain.

To make the real part of the average complex frequency response function estimates even, the functions are sampled up to the Nyquist frequency and the values are folded about the Nyquist frequency.

To make the imaginary part of the average complex frequency response function estimates odd, the imaginary parts are both folded and flipped about the Nyquist frequency.

By folding about the Nyquist frequency the mirroring step produces average complex frequency response function estimates the same length as the original time history sections. The functions are now ready for the final Have Derivatives process step, inverse transforming back into the time domain.

2.5.3 Inverse Transformation. It has been shown that the impulse response function of a linear system is related to the frequency response function of the system by the Fourier integral (Newland, 1993:60 to 62). A discrete approximation of this relationship is achieved by inverse fast Fourier transforming the frequency response function to produce a discrete pulse response function.

The last Have Derivatives step begins by inverse fast Fourier transforming the full average complex frequency response function estimates back into the time domain to give the aircraft's responses to a discrete pulse input. The discrete pulse responses are multiplied by the sampling

rate in samples per second to give a correct approximation to the continuous inverse Fourier transform results (Brigham, 1988:197).

A set of aircraft responses and the corresponding input are required to estimate stability derivatives using PCPID. Because the inverse transform produces only the responses, an appropriate input is simulated as part of the Have Derivatives process to be saved with the responses for PCPID processing. A simulated ideal impulse, one time step wide and one over one time step in amplitude was found to be an appropriate input. The justification for choosing this input will be discussed in Chapter 4, Simulation Results, and Chapter 6, Flight Test Results.

The Have Derivatives process was incorporated into a simulation program that will be presented in Chapter 3, Simulation Development. The process was also incorporated as part of the flight test data reduction routine discussed in Chapter 5, Flight Test Development.

2.6 Summary

The theoretical background necessary to understanding information presented in subsequent chapters was presented in this chapter using a build-up approach.

In particular, the output-error parameter estimation method was examined in the first main subsection through a description of the main components of the method as they are implemented in PCPID.

The five subroutines that comprise PCPID were presented in the next subsection.

Ensemble averaging was developed in subsection 2.4 as a technique to minimize the random variations caused by process and measurement noise while preserving the true system dynamics. The motivation for using this technique was the inability of the output-error parameter estimation method to account for process noise or measurement noise on the input signal.

The final main subsection presented the complete Have Derivatives process. The process can be broken into three interrelated steps.

Input and response time histories are measured and ensemble averaged using Welch's method. The average complex frequency response function for each input and response combination is estimated from the average spectral density estimates.

Full average complex frequency response function estimates are created by folding the real components and by folding and flipping the imaginary components of the functions about the Nyquist frequency.

The full average complex frequency response function estimates are inverse fast Fourier transformed back into the time domain as discrete pulse responses. A simulated ideal impulse input time history is created to correlate with the discrete pulse responses. The discrete pulse responses and simulated ideal impulse input are saved together for PCPID processing.

The Have Derivatives process was incorporated into both a simulation program and a flight test data reduction program. The simulation program will be described completely in the next chapter, Simulation Development. The description and implementation of the flight test data reduction program are presented in Chapter 5, Flight Test Development.

3. Simulation Development

3.1 Overview

The program used to evaluate the Have Derivatives process in simulation is developed in this chapter. A build-up approach is followed as before, starting with definitions of the sign and units convention, continuing with development of the state space model used in the simulation program, and concluding with a description of the complete program.

The sign and units convention used in simulation and flight test during this research are graphically presented in the first subsection.

The basic model is developed next from aircraft short period approximation equations of motion. The equations are presented along with assumptions made in their derivation.

The modeling of process noise as turbulence, using the Dryden form of the spectra for vertical disturbance, is discussed in the chapter's second main subsection. Variable process noise is added to the model through the aerodynamic terms of the basic equations of motion.

Next, variable measurement noise is modeled by passing pseudo-white noise through a high-pass Butterworth filter and then scaling the noise appropriately for each input and response signal. Measurement noise is added to the input and response signals before saving them for parameter estimation processing.

In the final main subsection the process and measurement noise models are incorporated into the basic aircraft state space model. The complete state space model is presented along with a description of its integration into the simulation program HD_SIM.M. Finally, the various operations performed by HD_SIM.M are described.

3.2 Sign and Units Convention

The sign convention used throughout this research is shown in Figure 3.1. Forces, moments and angular rates were all referenced to the aircraft body axes, which have their origin at the aircraft center of gravity. This is the convention followed by the USAF Test Pilot School (TPS) and the Personal Computer Parameter Identification (PCPID) software used in this research.

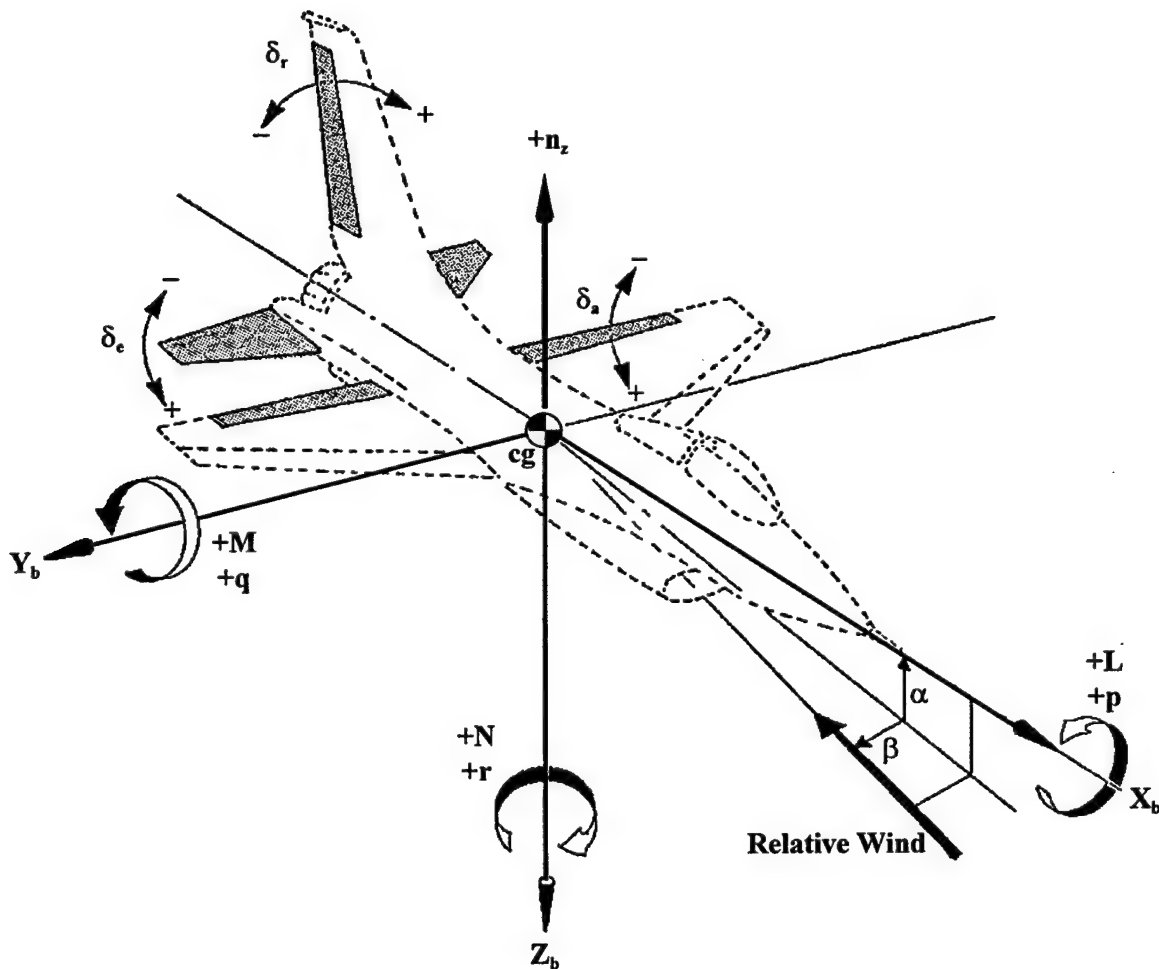


Figure 3.1. Simulation and Flight Test Sign Convention

To be consistent with USAF TPS data processing convention and parameter estimation software, English units were used throughout simulation and flight test.

3.3 Basic Model Development

Due to limited flight test time and resources, derivatives from only one axis of motion were evaluated to satisfy the research objectives stated in Chapter 1. It did not matter which axis was chosen as long as the same axis and its associated derivatives were studied throughout. This was because the evaluations were based on relative comparisons of results from subsequent simulation and flight test runs to their respective basis results. The longitudinal axis was chosen because it was the simplest to model and analyze.

The basic state space model was developed from the two degree of freedom longitudinal equations of motion given as Equations (3.1a) and (3.1b).

$$\dot{\alpha} = q - \frac{\bar{q} S}{m V_t} (C_{N\alpha} \alpha + C_{N\delta e} \delta e) - \frac{g \alpha_0}{V_t} \alpha \quad (3.1a)$$

$$\dot{q} = \frac{\bar{q} S c}{I_{yy}} (C_{m\alpha} \alpha + \frac{c}{2 V_t} C_{mq} q + C_{m\delta e} \delta e) \quad (3.1b)$$

where

α	=	angle of attack	(deg)
α_0	=	angle of attack at time $t = 0$	(deg)
q	=	pitch rate	(deg / sec)
$\dot{\alpha}$	=	derivative of angle of attack with respect to time	(deg / sec)
\dot{q}	=	derivative of pitch rate with respect to time	(deg / sec)
\bar{q}	=	dynamic pressure	(lb _f / ft ²)
V_t	=	true velocity	(ft / sec)
g	=	acceleration due to gravity (32.174)	(ft / sec ²)

m	=	aircraft mass	(slugs)
I_{yy}	=	aircraft moment of inertia about the Y_t axis	(slugs / ft ²)
S	=	wing area	(ft ²)
c	=	aircraft reference chord	(ft)
δe	=	elevator or horizontal stabilator deflection	(deg)
$C_{N\alpha}$	=	coefficient of normal force due to angle of attack	(per deg)
$C_{N\delta e}$	=	coefficient of normal force due to elevator control	(per deg)
$C_{m\alpha}$	=	coefficient of pitching moment due to angle of attack	(per deg)
C_{mq}	=	coefficient of pitching moment due to pitch rate	(per rad)
$C_{m\delta e}$	=	coefficient of pitching moment due to elevator control	(per deg)

Equations (3.1a) and (3.1b), commonly referred to as the short period approximation equations of motion, were developed following the treatment found in the USAF TPS Flying Qualities Phase Text with two notable exceptions (Flying Qualities Branch, 1992:1 to 136). First, angle of attack was used as an approximation for pitch angle in the gravity term of Equation (3.1a). The pitch angle equation was omitted in the development of the short period approximation equations because it added little information while making the simulations more complicated (Maine and Iliff, 1986:28). Although the pitch angle does change during longitudinal maneuvers, the effect of the motion on the short period dynamics is small. Therefore, the gravity term in Equation (3.1a) is well approximated by setting pitch angle equal to angle of attack (Maine and Iliff, 1986:23).

The only other difference from the derivation in the text was the combining of $C_{m\dot{\alpha}}$ with C_{mq} . This is common practice in flight test because the results from estimating the derivatives separately have been poor while the results from estimating a combined pitch damping derivative

have been much better (Nagy, 1976:17; Maine and Iliff, 1986:4 to 5; Flying Qualities Branch, 1992:145).

Several assumptions were made in the derivation of Equations (3.1a) and (3.1b) that are worth repeating (Flying Qualities Branch, 1992:1 to 136).

1. The aircraft has constant mass during the maneuver.
2. Propulsion and rotating mass effects are negligible during the maneuver.
3. The aircraft is a rigid body.
4. A flat-earth inertial reference frame is suitably inertial during the maneuver.
5. The atmosphere is at rest relative to a flat-earth inertial frame.
6. Gravitational acceleration is constant during the maneuver.
7. The $X_b Y_b$ plane of the aircraft is a plane of symmetry, with $I_{xy} = I_{yz} = 0$.
8. True airspeed is constant during the maneuver.
9. The maneuver causes only small perturbations about a trimmed flight condition with angle of attack remaining less than 20° .
10. The trimmed condition is straight and level flight.
11. The X_b axis speed component is constant during the maneuver.

These assumptions had no impact on the research findings because the same model was used for all simulations, with results then compared relative to each other.

The basic state space model, formed from Equations (3.1a) and (3.1b), are presented as Equations (3.2a) and (3.2b).

$$\dot{x} = Ax + Bu \quad (3.2a)$$

$$y = Cx + Du \quad (3.2b)$$

where

$$\dot{x} = \begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} \quad x = \begin{bmatrix} \alpha \\ q \end{bmatrix} \quad u = [\delta_e] \quad y = \begin{bmatrix} \alpha \\ q \\ n_z \end{bmatrix}$$

$$A = \begin{bmatrix} -\frac{\bar{q} S}{m V_t} C_{N\alpha} - \frac{g \alpha_0}{V_t} & 1 \\ \frac{\bar{q} S c}{I_{yy}} C_{m\alpha} & \frac{\bar{q} S c^2}{I_{yy} 2 V_t} C_{mq} \end{bmatrix} \quad B = \begin{bmatrix} -\frac{\bar{q} S}{m V_t} C_{N\delta_e} \\ \frac{\bar{q} S c}{I_{yy}} C_{m\delta_e} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & -\frac{(x_{cg} - x_\alpha)}{V_t} \\ 0 & 1 \\ \frac{\bar{q} S}{m g} C_{N\alpha} + \frac{(x_{cg} - x_{nz})}{g} \frac{\bar{q} S c}{I_{yy}} C_{m\alpha} & \frac{(x_{cg} - x_{nz})}{g R} \frac{\bar{q} S c^2}{I_{yy} 2 V_t} C_{mq} \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ \frac{\bar{q} S}{m g} C_{N\delta_e} + \frac{(x_{cg} - x_{nz})}{g} \frac{\bar{q} S c}{I_{yy}} C_{m\delta_e} \end{bmatrix}$$

x_{cg} = center of gravity distance from reference along X_b (positive aft) (ft)

x_α = angle of attack sensor distance from reference along X_b (positive aft) (ft)

x_{nz} = normal accelerometer distance from reference along X_b (positive aft) (ft)

R = conversion factor (~ 57.3) (deg)

3.4 Process Noise Model Development

Because the output-error parameter estimation method does not account for process noise, the main focus of the research was to evaluate the Have Derivatives process to reduce errors caused by process noise. Turbulence is the standard example of process noise (Iliff, 1987:4).

Guidance found in MIL-STD-1797A, APPENDIX A, was used to model turbulence. The Dryden form of turbulence spectra was used because no comparable structural analyses were to be accomplished in this research (Department of Defense, 1995:657). To first order, only turbulence along the Z_0 axis affected the short period responses through the equations of motion so only the vertical component of turbulence was developed.

The development began with the Dryden form of the vertical component turbulence spectra as given in MIL-STD-1797A, APPENDIX A (Department of Defense, 1995:657). A linear angle of attack disturbance filter was developed to take white noise as an input and produce a signal with the same power spectral density as the Dryden spectra. The output from the angle of attack disturbance filter, when divided by the average forward velocity of the aircraft, gave the incremental change in angle of attack due to turbulence.

In a similar manner, a pitch rate disturbance filter was developed that took the output from the angle of attack disturbance filter as an input and produced the incremental change in pitch rate due to turbulence. Detailed derivation of turbulence shaping filters can be found in references (Stevens and Lewis, 1992:503 to 504; Roskam, 1979:867 to 897).

The angle of attack and pitch rate disturbance filters were combined in state space form as shown in Equations (3.3a) and (3.3b).

$$\dot{x}_d = A_d x_d + B_d \eta \quad (3.3a)$$

$$y_d = C_d x_d + D_d \eta \quad (3.3b)$$

where

$$\dot{x}_d = \begin{bmatrix} \dot{\alpha}_{d1} \\ \dot{\alpha}_{d2} \\ \dot{q}_d \end{bmatrix} \quad x_d = \begin{bmatrix} \alpha_{d1} \\ \alpha_{d2} \\ q_d \end{bmatrix} \quad y_d = \begin{bmatrix} \alpha_d \\ q_d \end{bmatrix}$$

η = zero mean, Gaussian distributed, band-limited white noise

$$A_d = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{V_t^2}{4L^2} & -\frac{V_t}{L} & 0 \\ -\frac{R\sigma\sqrt{V_t}}{L\sqrt{8\pi L}} & -\frac{R\sigma\sqrt{3}}{\sqrt{2\pi L V_t}} & -\frac{R V_t \pi}{4b} \end{bmatrix} \quad B_d = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C_d = \begin{bmatrix} -\frac{R\sigma\sqrt{V_t}}{L\sqrt{8\pi L}} & -\frac{R\sigma\sqrt{3}}{\sqrt{2\pi L V_t}} & 0 \\ -\frac{R\sigma\sqrt{V_t^3\pi}}{8bL\sqrt{2L}} & -\frac{R\sigma\sqrt{3V_t\pi}}{4b\sqrt{2L}} & -\frac{R V_t^2 \pi^2}{16b^2} \end{bmatrix} \quad D_d = [0]$$

V_t = aircraft true velocity (ft / sec)

L = turbulence scale length (ft)

σ = root-mean-square (rms) turbulence intensity (ft / sec)

b = wing span (ft)

A turbulence scale length of 875 feet was chosen for the AT-37B simulations to model assumed isotropic turbulence found higher than 2,000 feet above ground level (Department of Defense, 1995:658 to 661). Turbulence level was set in simulations by adjusting the root-mean-square turbulence intensity term inside the simulation program.

3.5 Measurement Noise Model Development

Measurement noise was simulated by passing zero mean, Gaussian distributed, band-limited white noise through a high-pass Butterworth filter and then multiplying the high frequency noise by a different gain for each signal.

A 4th order Butterworth filter was created with a cutoff frequency of 100 radians per second, well above any dynamics of interest. This would help to discern the degrading effects due to the high frequency simulated measurement noise from the estimation errors due to the low frequency simulated process noise. The 100 radian per second cutoff frequency was approximately 50 percent of the Nyquist frequency with the 67 samples per second sampling rate used in simulation and flight test. This cutoff frequency provided a good simulation of actual measurement noise which is generally minimized with a low-pass anti-aliasing filter at 40 percent of the Nyquist sampling frequency (Maine and Iliff, 1986:104).

The gain for each signal was chosen based on historical measurement uncertainty of USAF TPS instrumentation. The measurement uncertainties for the AT-37B and F-16B are given in Table 3.1.

Table 3.1. AT-37B and F-16B Approximate Measurement Uncertainties

Parameter	AT-37B ¹ Measurement Uncertainty	F-16B ² Measurement Uncertainty
Angle of Attack (deg)	0.15	0.20
Pitch Rate (deg / sec)	1.50	0.10
Normal Acceleration (g)	0.15	0.04
Elevator/Stabilator Deflection (deg)	0.60	0.15

¹(Instrumentation Branch, 1996a:1 to 2)

²(Instrumentation Branch, 1996b:4-15 to 4-17)

The measurement noise for each parameter was scaled by multiplying the output of the Butterworth filter by a percentage of the associated measurement uncertainty. The measurement noise level was controlled by adjusting the percentage in the simulation program.

Figure 3.2 illustrates the measurement noise model.

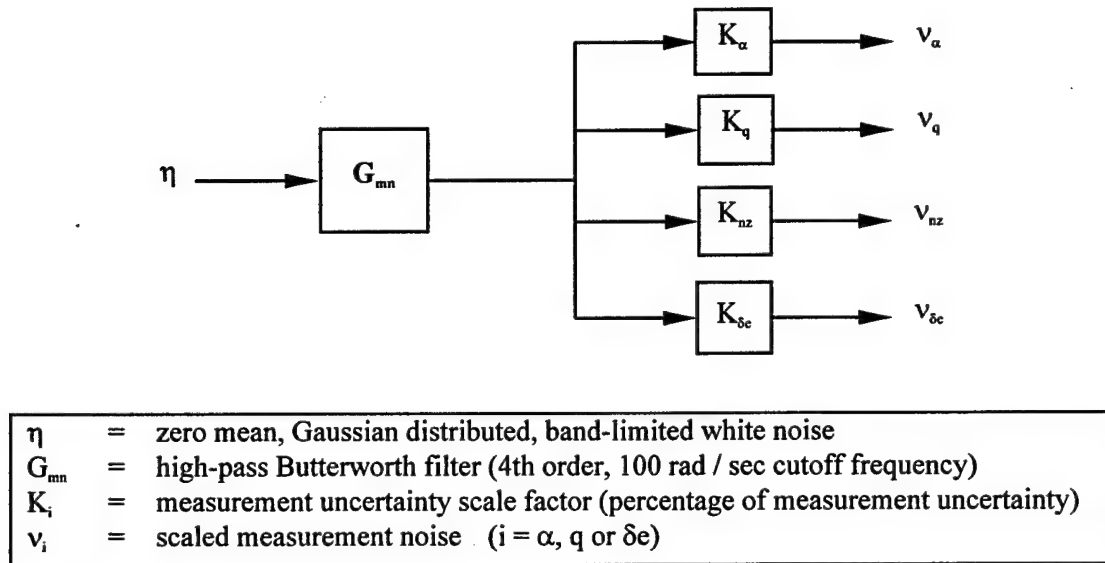


Figure 3.2. Measurement Noise Model

The scaled measurement noise for angle of attack, pitch rate and normal acceleration is added to the associated signal in the output equation of the complete state space model that is developed in the next chapter subsection. The scaled measurement noise for elevator or stabilator deflection is added to that signal separately in the simulation program, after the input signal has been used in the linear simulation and before it is saved with the response signals for PCPID processing.

3.6 Complete Model and Simulation Program

The basic equations, augmented with process and measurement noise terms, were formed into a linear, continuous, time-invariant state space model given as Equations (3.4a) and (3.4b).

$$\begin{bmatrix} \dot{x} \\ \dot{x}_d \end{bmatrix} = \begin{bmatrix} A & GC_d \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x \\ x_d \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & B_d \end{bmatrix} \begin{bmatrix} \delta_e \\ \eta \\ v \end{bmatrix} \quad (3.4a)$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_d \end{bmatrix} + \begin{bmatrix} D & 0 & I \end{bmatrix} \begin{bmatrix} \delta_e \\ \eta \\ v \end{bmatrix} \quad (3.4b)$$

where

$$G = \begin{bmatrix} -\frac{\bar{q} S}{m V_t} C_{N\alpha} & 0 \\ \frac{\bar{q} S c}{I_{yy}} C_{m\alpha} & \frac{\bar{q} S c^2}{I_{yy} 2 V_t} C_{mq} \end{bmatrix} \quad v = \begin{bmatrix} v_\alpha \\ v_q \\ v_{\delta e} \end{bmatrix}$$

The **G** matrix was formed from the aerodynamic terms of the basic state space model **A** matrix. The process noise affected angle of attack and pitch rate response only through these terms. The **v** vector was the measurement noise signals produced by the model in Figure 3.2.

The complete model was coded into a MATLAB® script file entitled HD_SIM.M. AT-37B flight data were obtained from the USAF TPS to complete the model. These data included flight conditions, aircraft dimensions, aircraft mass distribution and stability derivative estimates from previous parameter estimation test flights.

In addition to the aircraft model and flight conditions, various elevator input functions were created in HD_SIM. The user could then select which input to use in a linear simulation. HD_SIM used the aircraft model along with the selected input in LSIM.M, a linear simulation utility in MATLAB's® Control System Toolbox® (Grace and others, 1992:2-106 to 2-107). The output from LSIM were angle of attack, pitch rate and normal acceleration time responses due to the specified input. Initial conditions were set to zero for all simulations. The input and response signals and flight conditions for the simulation run were saved for PCPID processing.

Process and measurement noise levels were individually varied as previously described. This allowed the user to run simulations with no noise, or at any noise level desired.

Also contained in HD_SIM was code to accomplish the Have Derivatives process described in Chapter 2. The code first took the input and response signals from the linear simulation and estimated average complex frequency response functions using the MATLAB® routine, TFE.M, from the Signal Processing Toolbox® (Krause and others, 1994:2-218 to 2-221). The ensemble averaging step of the Have Derivatives process was accomplished by TFE.M. Ensemble averaging variables, including section length, amount of overlap and the window function to be applied to each section before averaging, were user specified in HD_SIM and given to TFE.M as additional inputs. After invoking TFE.M, HD_SIM mirrored the average complex frequency response function estimates returned by TFE.M and transformed them back into the time domain as discrete pulse responses. Finally, HD_SIM saved the discrete pulse responses and a simulated ideal impulse input in the GetData format required for PCPID processing (Barnicki, 1996).

The last operation HD_SIM performed was to plot various time and frequency domain results including input and response time histories, frequency responses and coherence functions.

Complete HD_SIM.M code is found at Appendix A, Simulation Program Code.

3.7 Summary

The simulation program, HD_SIM.M, was developed in this chapter. The following steps were used in a build-up approach which started with the sign and units convention and concluded with the complete simulation program.

1. The sign and units convention used in simulation and flight test was defined.
2. The basic aircraft model was developed from the short period approximation equations of motion. Assumptions made in the development of the model were given.
3. The process noise model was developed from the Dryden vertical component turbulence spectra. Angle of attack and pitch rate noise filters were represented as a combined disturbance filter state space model. The filter took pseudo-white noise as an input and produced noise shaped according to the Dryden model. The process noise model was later incorporated into the basic aircraft model through the aerodynamic terms.
4. Measurement noise was simulated by passing pseudo-white noise through a high-pass Butterworth filter. The high frequency noise was then individually scaled and added to the input and response signals after the linear simulation and before saving them for PCPID processing.
5. The process noise and measurement noise models were incorporated into the basic aircraft state space model.
6. The various operations accomplished by the complete simulation program, HD_SIM.M, were described.

Simulation results are presented and analyzed in the next chapter. The results were produced by PCPID using simulated data from runs made with HD_SIM, the simulation program developed in this chapter.

4. Simulation Results and Analysis

4.1 Overview

This chapter presents representative results of the computer simulations run using the simulation program, HD_SIM.M, developed in the last chapter. These results, which include simulation output and stability derivative estimation results from PCPID, are analyzed and appropriate conclusions are drawn. Complete results can be found in Appendix B.

The first main subsection presents simulation results using the current method of parameter estimation at the USAF Test Pilot School. The input and response signals were imported directly into PCPID to estimate stability derivatives. Because the data were not operated on by the Have Derivatives process, no ensemble averaging was accomplished. Runs were performed with no noise and with various levels of process and/or measurement noise using a doublet input. No noise simulations were also accomplished with input functions other than a doublet.

The results of the simulations that incorporated ensemble averaging are presented in the second main subsection of this chapter. A broadband input was used for all of these runs. The input and response signals were pre-processed using the Have Derivatives process as described in Chapter 2. The discrete pulse responses produced by the Have Derivatives process were imported into PCPID along with a simulated ideal impulse elevator input. This input correlated with the discrete pulse responses, allowing PCPID to estimate stability derivatives. Simulations were performed using a parametric analysis methodology to assess the effects of changing each of the ensemble averaging variables discussed in Chapter 2. Stability derivative estimation results from simulations where overlap length, section length, and maneuver length were varied are compared to simulation baseline results reported in subsection 4.2.

4.2 Simulation Results From PCPID Alone

Simulations were accomplished to examine the degrading effects of process and measurement noise on parameter estimation results and to provide a basis for comparison when evaluating the effectiveness of the Have Derivatives process in minimizing those effects.

Because the stability derivative values given by PCPID are estimates, it is not possible to calculate their absolute accuracy. In this research, however, that was not a hindrance as absolute accuracy was not necessary, but rather, the relative accuracy of the estimates as compared with a common benchmark across runs.

The relative accuracy was assessed by differencing the stability derivatives estimated from a given simulation with the actual values of those derivatives used to build the aircraft simulation model. A weighted root-sum-of-squares (rss) of those differences was then used to compare one run versus another.

The stability derivative weights used in the rss calculations were based on the derivatives' relative importance in determining the dynamic characteristics of an aircraft. Composite weights were derived from the three references shown in Table 4.1.

Table 4.1. Stability Derivative Weights

Derivative	Roskam ¹	FQ Branch ²	Nagy ³	Composite
$C_{N\alpha}$ (per deg)	10	"Strong"	8-9	9
$C_{N\delta e}$ (per deg)	not rated	"Moderate"	2-3	4
$C_{m\alpha}$ (per deg)	10	"Very Strong"	7-8	9
C_{mq} (per rad)	9	"Moderate"	1-2	5
$C_{m\delta e}$ (per deg)	not rated	"Very Strong"	10	10

¹(Roskam, 1982:235 to 236)

²(Flying Qualities Branch, 1992:302)

³(Nagy, 1997)

The root-sum-of-squares magnitude for each run was calculated from Equation (4.1).

$$\text{RSS Magnitude} = \{ \sum [\text{derivative weighting} * (\text{estimate value} - \text{actual value})^2] \}^{\frac{1}{2}} \quad (4.1)$$

where the summation was over the five derivatives.

4.2.1 Process Noise Only Simulation Results. Process noise was simulated as turbulence from the Dryden form as discussed in Chapter 3. Process noise level was adjusted through the rms vertical turbulence intensity term. To determine the effects of increasing process noise on stability derivative estimation results, simulations were accomplished using a doublet input and zero measurement noise at rms turbulence intensities of 3, 6 and 9 ft/sec. These intensities roughly correspond to light, moderate and heavy turbulence levels (Hoh and others, 1982:712).

For purposes of this research, the rms vertical turbulence intensity of 3 ft/sec was defined as calm air while 9 ft/sec was defined to be turbulent air. While it is possible that a flight could encounter occasional 6 ft/sec rms vertical turbulence intensities and still accomplish many of its objectives, it is unlikely that the same flight would begin or continue if 9 ft/sec turbulence was forecast or encountered. However, in order to discern the degrading effect of increasing process noise and the effectiveness of the Have Derivatives process at minimizing those degradations, it was necessary to select cases that would provide the required contrast. Therefore only rms vertical turbulence intensities of 3 and 9 ft/sec were used to evaluate the effectiveness of the Have Derivatives process in later simulations.

The impact of increasing process noise on aircraft response can be seen in Figures 4.1 and 4.2. The ideal response in these figures corresponds to an elevator doublet control input of +/- 1 degree deflection and no noise while the measured time histories include both the aircraft response to the elevator doublet and the variations due to process noise.

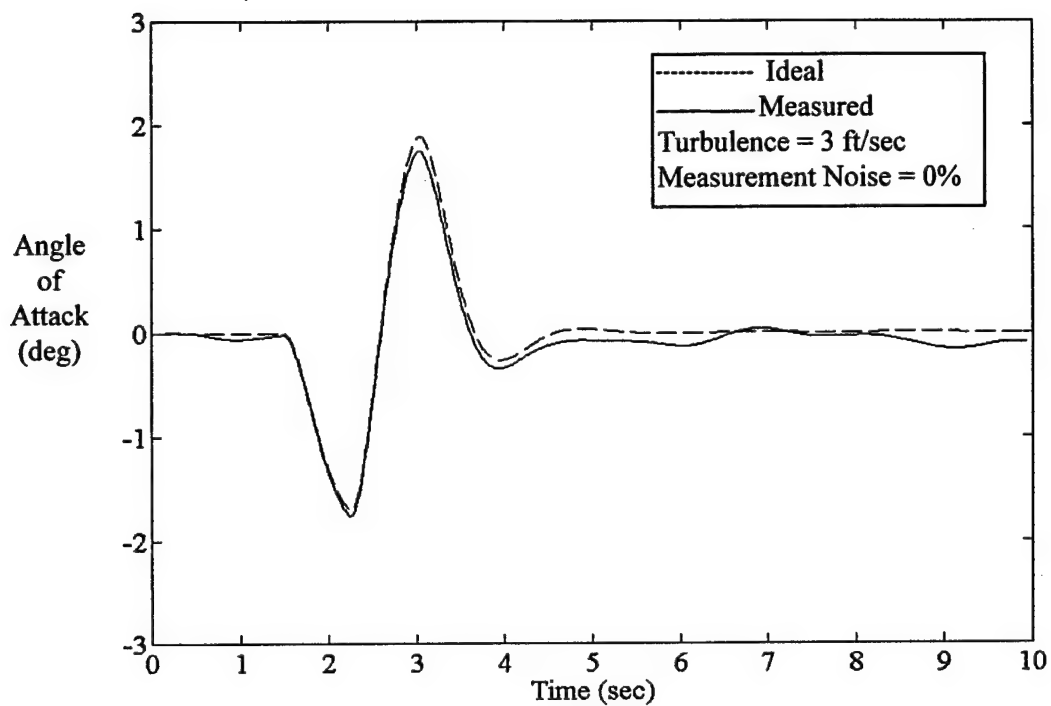


Figure 4.1. Doublet Responses, Ideal and Measured with 3 ft/sec Turbulence Intensity

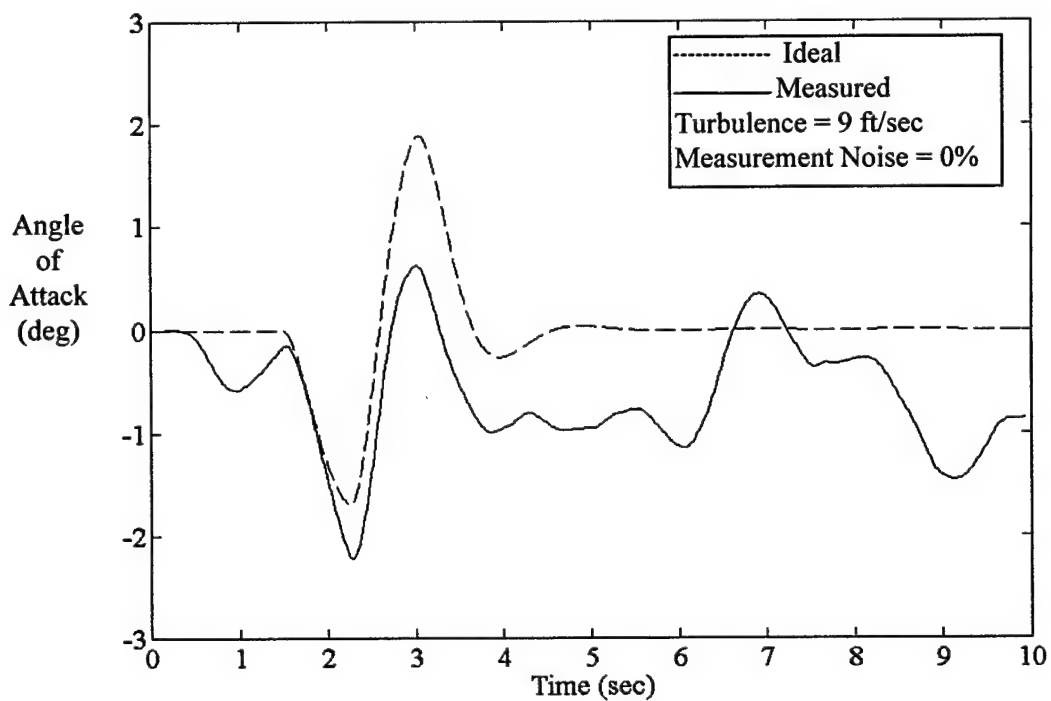


Figure 4.2. Doublet Responses, Ideal and Measured with 9 ft/sec Turbulence Intensity

The simulated measured time response deviates more from the ideal (no noise) time response as the rms vertical turbulence intensity increases. The consequence of increasing process noise is greater stability derivative estimate deviations, as shown in Figure 4.3.

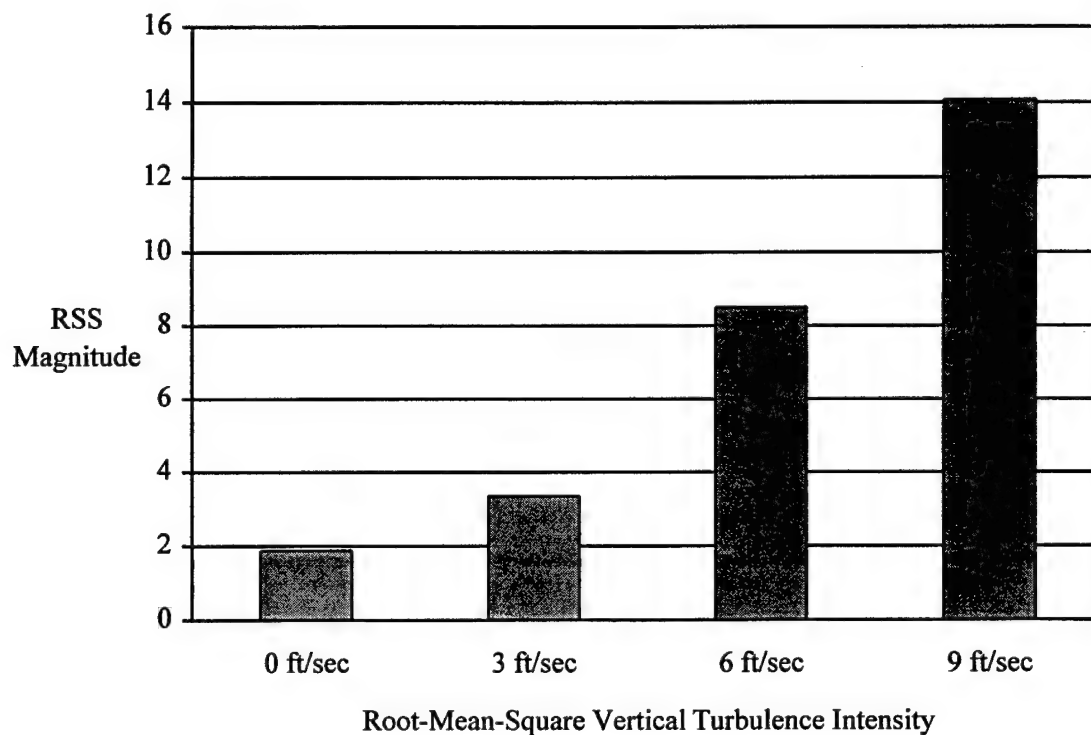


Figure 4.3. Root-Sum-of-Squares Magnitude versus Root-Mean-Square Turbulence Intensity

Notice that there were stability derivative deviations for even the ideal case with no noise. This was due to the fact that the derivative values produced by PCPID are estimates based on the match between the computed and measured time histories as discussed in Chapter 2. The slight deviations are a byproduct of the parameter estimation process. The important result for the purposes of this research is not the absolute derivative accuracy, but rather, the relative comparison between runs.

4.2.2 Measurement Noise Only Simulation Results. In a manner similar to the process noise only simulations, runs were made to assess the effects of increasing measurement noise on stability derivative estimate accuracy. A ± 1 degree elevator deflection doublet input was again used for these runs. Root-mean-square turbulence intensity was fixed at 0 ft/sec, to isolate the effects due to measurement noise.

Measurement noise was simulated as described in subsection 3.5 of Chapter 3. The measurement noise level in these simulations was defined by the percentage of measurement uncertainty used to scale the high frequency noise. For the purposes of this research, a measurement noise level of 10% was defined as calm while 30% was considered a turbulent noise level that would likely contribute to degraded results. These definitions were chosen to coincide with the calm and turbulent air conditions described earlier for process noise. A 20% measurement noise level was used in the baseline runs to quantify the effects of measurement noise on stability derivative estimate accuracy but was not used in the evaluations of the Have Derivatives process in subsequent simulations.

The effects of increasing measurement noise on the angle of attack time history measurements can be seen in Figures 4.4 and 4.5 on the following page. The ideal response in these figures corresponds to an elevator doublet control input of ± 1 degree deflection and no noise while the measured time histories include both the aircraft response to the elevator doublet and the variations due to measurement noise.

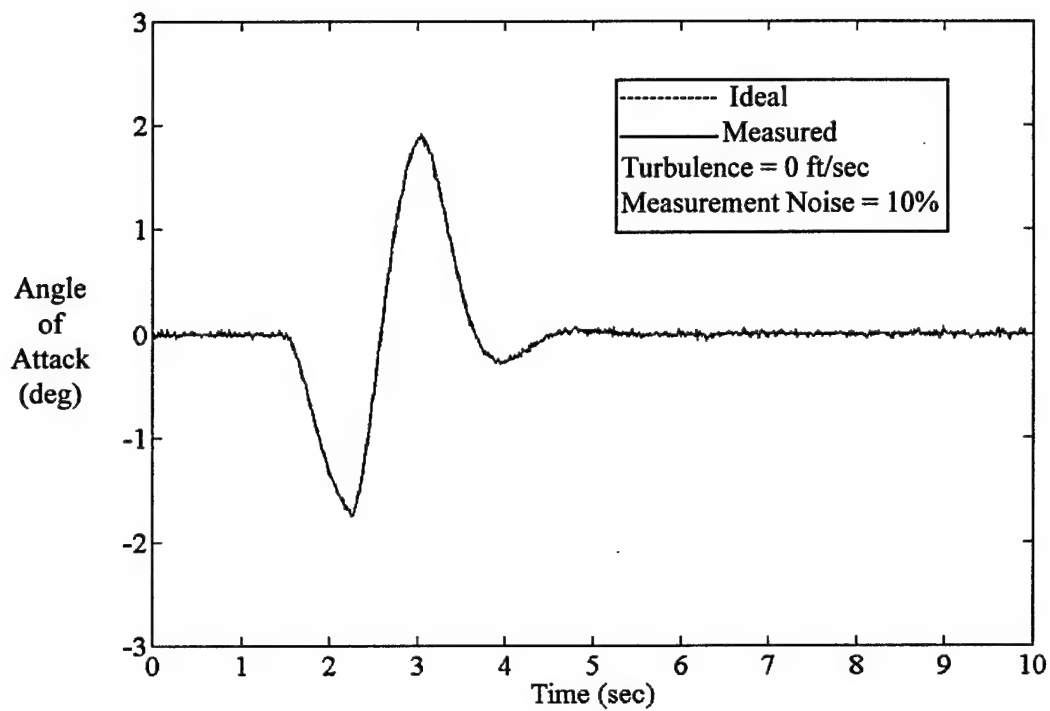


Figure 4.4. Doublet Responses, Ideal and Measured with 10% Measurement Noise

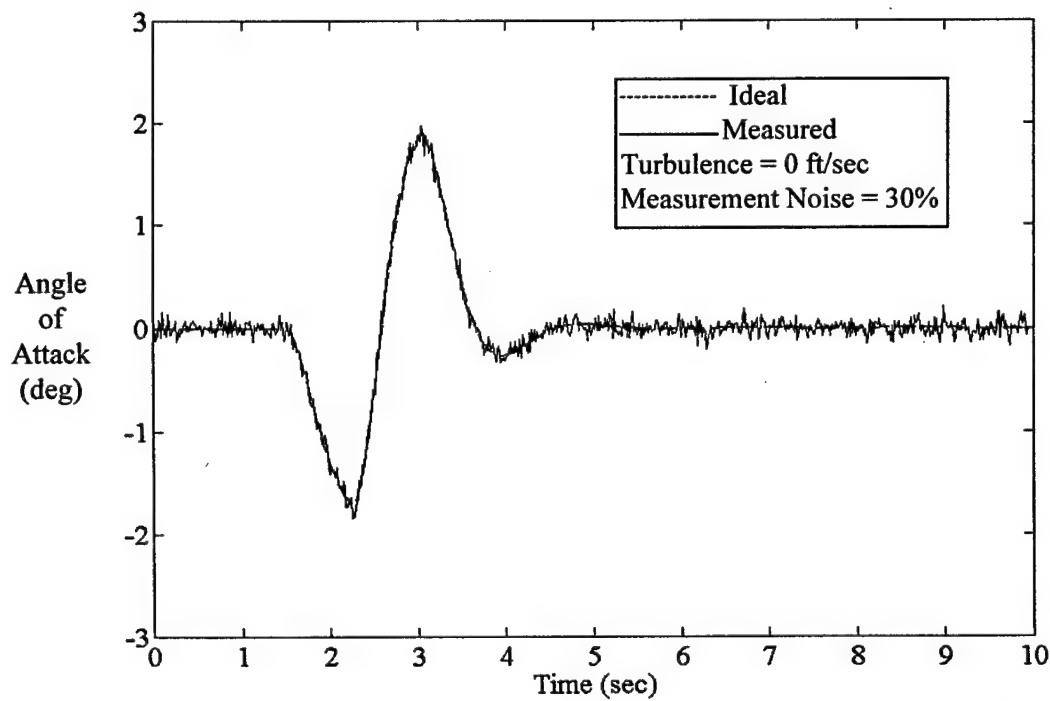


Figure 4.5. Doublet Responses, Ideal and Measured with 30% Measurement Noise

Although it appears that the measurement noise had little effect on the angle of attack time histories, the effect on stability derivative estimate accuracy was as degrading as from increasing rms turbulence intensity. This is illustrated in Figure 4.6.

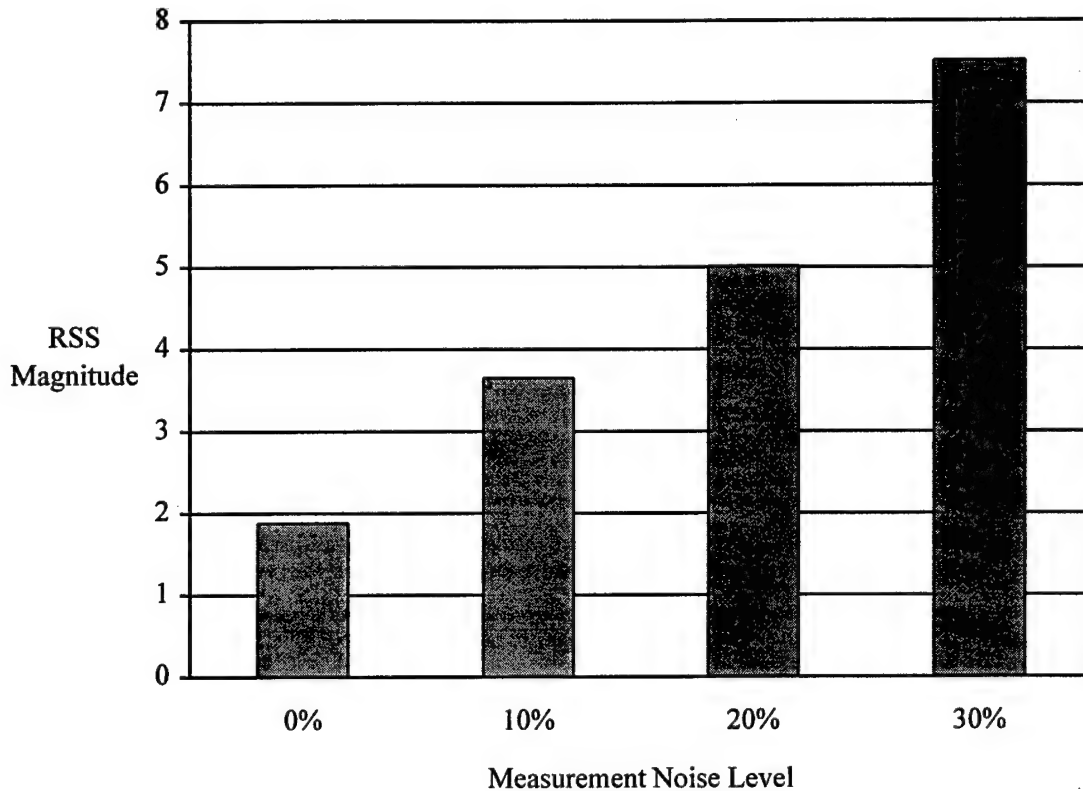


Figure 4.6. Root-Sum-of-Squares Magnitude versus Measurement Noise Level

Besides the deviations due to uncorrelated random variations on the response signals, an additional cause of the increasing deviations can be found in Figure 2.1 of Chapter 2. The output-error parameter estimation method implemented in PCPID does not account for measurement noise on the control input signal. The parameter estimation program had difficulty correlating the responses to the input, resulting in poorer time history matches and greater stability derivative deviations.

4.2.3 Combined Process and Measurement Noise Simulation Results. In addition to the runs made where rms turbulence intensity or measurement noise level were varied individually, simulations were accomplished with the same elevator doublet input as before, but with various combined rms turbulence intensity and measurement noise level settings.

The first run was made with rms vertical turbulence intensity set to 0 ft/sec and measurement noise level set to 0%. These were defined as ideal conditions with the PCPID stability derivative estimation results from this run representing a theoretical best possible from the current flight test techniques and data processing methods.

The next simulation had rms turbulence intensity set at 3 ft/sec and measurement noise level set to 10%. As mentioned previously, these conditions were defined as what might be encountered on a typical parameter estimation test flight in calm air.

The final combined conditions simulation was made using what was defined as turbulent conditions. Root-mean-square turbulence intensity was set to 9 ft/sec and a measurement noise level of 30% was used.

The runs with calm and turbulent conditions demonstrated the capabilities of PCPID to handle degraded flight test data and provided a basis for comparing the subsequent results when the Have Derivatives process was incorporated.

Figures 4.7 and 4.8 illustrate a representative time history as affected by calm and turbulent combined conditions respectively. In both figures the dashed line is the ideal conditions time history (no process or measurement noise) while the solid line shows the angle of attack time history response as degraded by the indicated combined conditions.

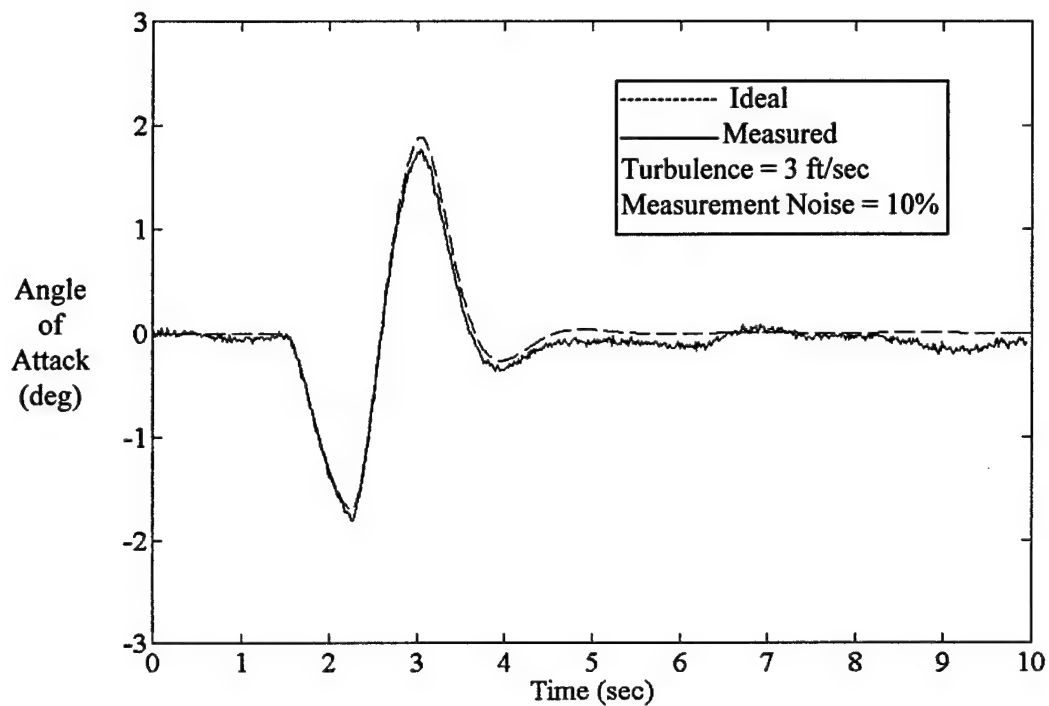


Figure 4.7. Doublet Responses, Ideal and Measured with Calm Conditions

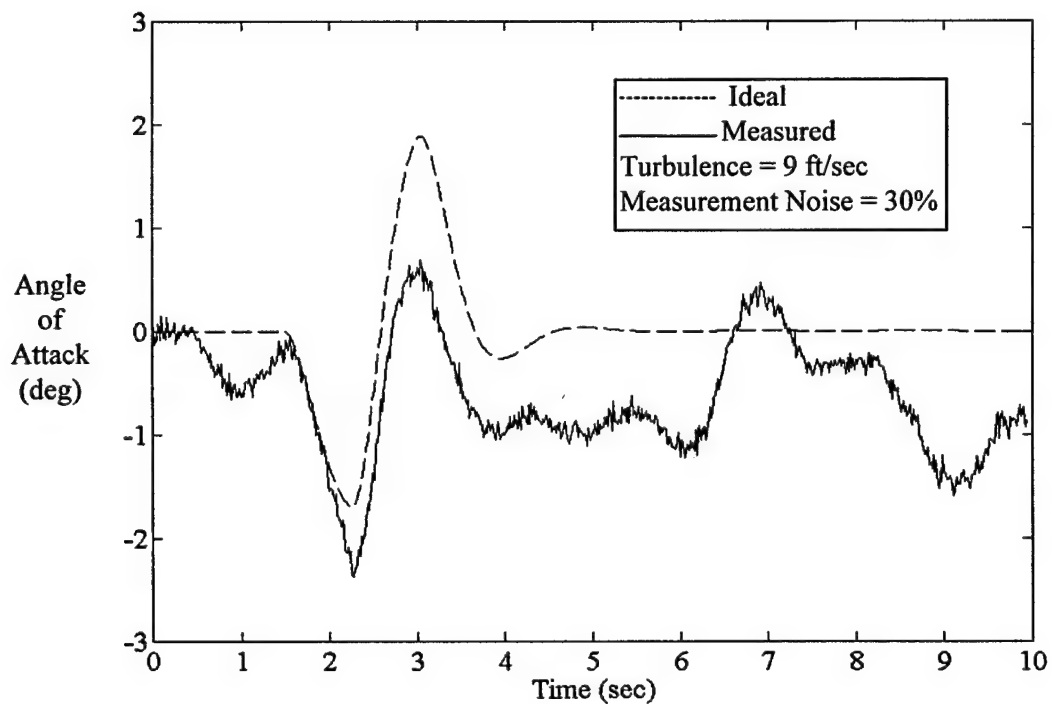


Figure 4.8. Doublet Responses, Ideal and Measured with Turbulent Conditions

The results of the combined conditions simulations are presented in Figure 4.9.

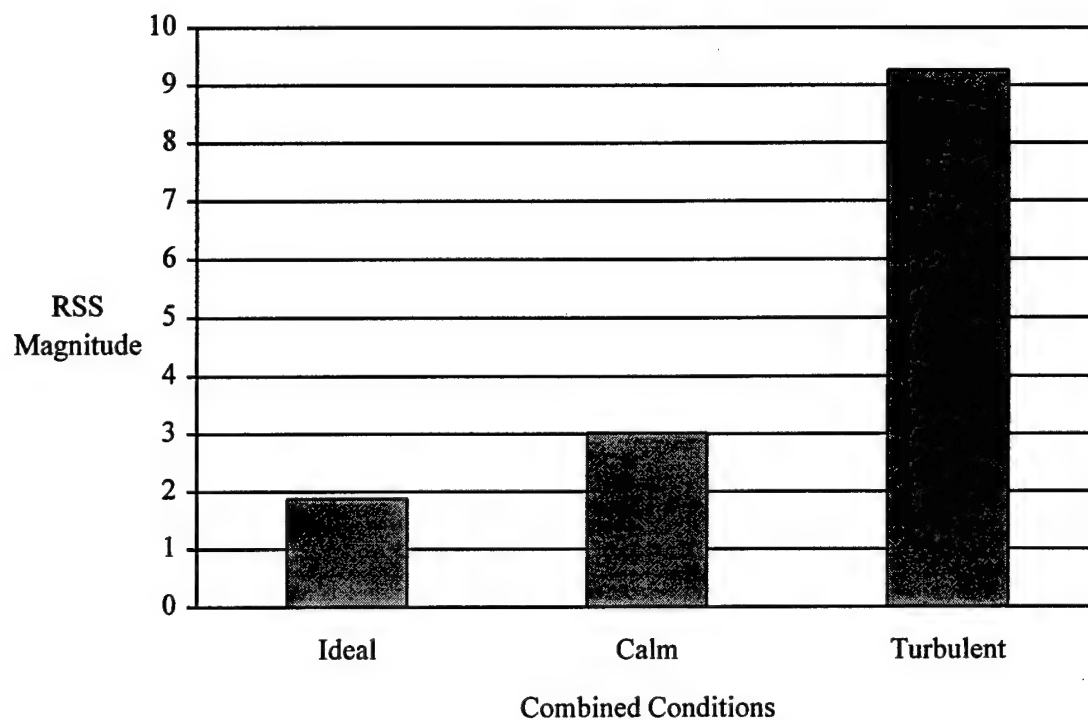


Figure 4.9. Root-Sum-of-Squares Magnitude versus Combined Conditions

Again, as expected, the stability derivative errors, estimate deviations from actual values, grew as conditions worsened.

4.2.4 No Noise Simulation Results Using Different Input Functions. The last set of PCPID only simulations were accomplished using ideal conditions and elevator inputs other than the standard doublet. While not specifically investigating the concept of an optimum input, these runs were made to assess the effects that different inputs had on stability derivative estimate accuracy. The input amplitudes were ± 1 degree elevator deflection except for the simulated ideal impulse.

The simulated ideal impulse was constructed one time step wide and one-over-one-time-step in amplitude to have unity area according to linear systems theory (Brigham, 1988:386 to 393). This equated to a signal 0.015 seconds wide and 67 degrees in amplitude based on the 67 samples per second rate used in all of the simulations. The time histories of the responses from a linear simulation using the simulated ideal impulse were nearly identical to the theoretical ideal impulse responses. This validated the simulated ideal impulse as a good approximation of the theoretical ideal impulse.

Figure 4.10 compares the different inputs to the doublet that is commonly used in flight test.

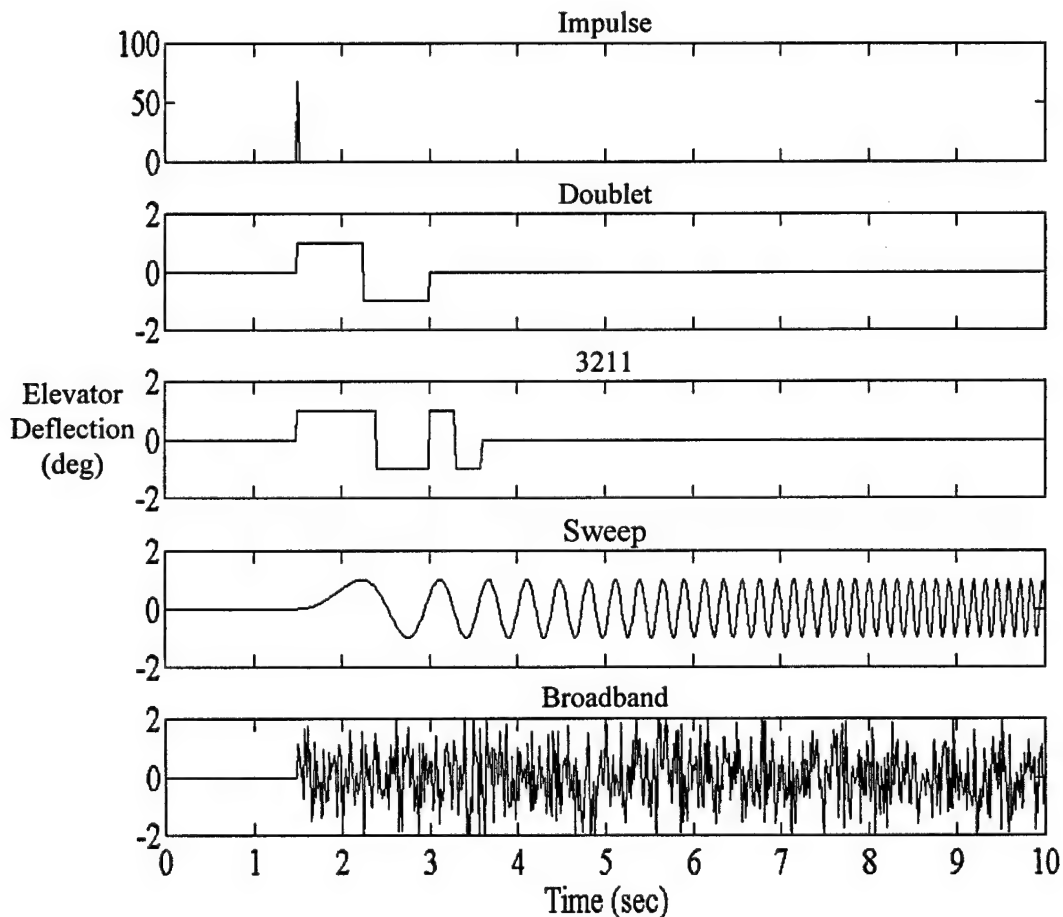


Figure 4.10. Elevator Deflection versus Time for Various Input Functions

The surprising results of these simulations are presented in Figure 4.11.

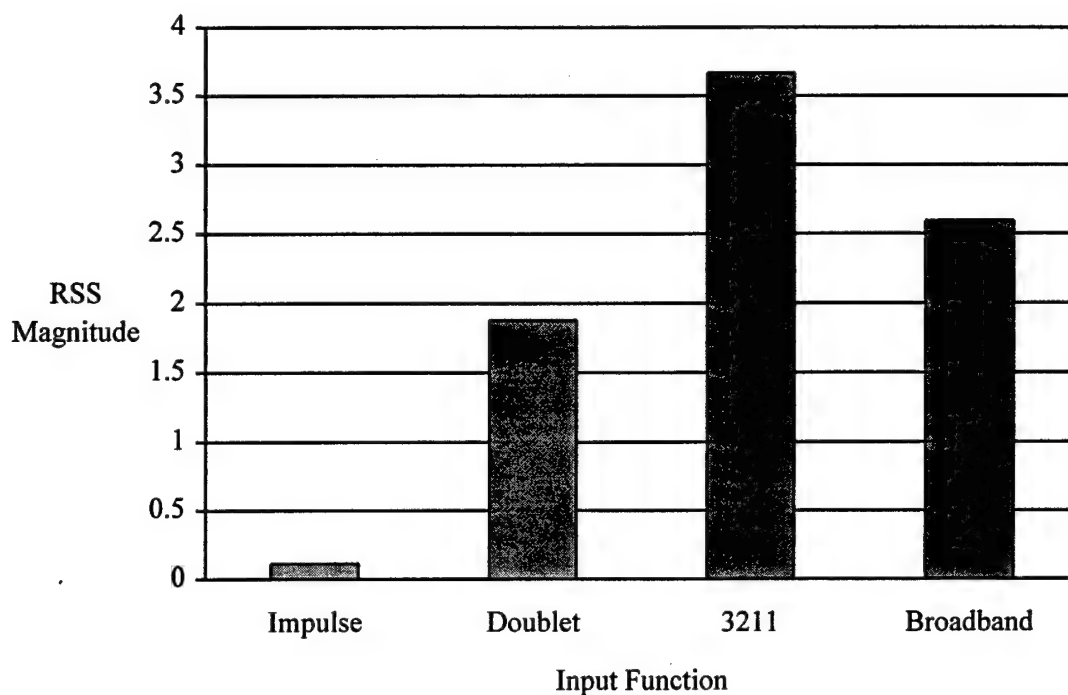


Figure 4.11. Root-Sum-of-Squares Magnitude versus Input Function

The real shock about these results was that while the standard doublet gave better results than the 3211 and broadband, the impulse input resulted in significantly better estimates, nearly identical to the actual values of the derivatives! The only problem is that a true impulse is impossible to produce in flight test due to lags and rate limiting in the flight control system.

The closest input to an impulse in flight test is a pulse. The problem with a pulse is that the resulting response is usually moving the aircraft away from the trim flight condition, possibly violating maneuver tolerances. Also, linearity assumptions could be violated by using a pulse.

Doublets are typically used in flight test because they are easy to accomplish and because the resulting response from the doublet input is usually symmetric about the trim flight condition.

This makes it easier for the pilot to remain within the maneuver tolerance and simplifies subsequent analysis as linearity assumptions are not violated.

The excellent results for the simulated ideal impulse input were exciting because, as explained in Chapter 2, the Have Derivatives process returns the discrete pulse response when the average frequency response function estimates are transformed back into the time domain. When plotted on the same figure, the discrete pulse responses were nearly identical to the impulse responses produced by the simulated ideal impulse input of Figure 4.11. Therefore, it was reasonable to expect that stability derivative estimation results, similar to those found for the simulated ideal impulse input of Figure 4.11, could be achieved by importing the discrete pulse response time histories into PCPID along with the simulated ideal impulse input time history.

4.3 Simulation Results From The Have Derivatives Process With PCPID

Although a doublet is commonly used as the stability derivative estimation flight test input, it was found that the best discrete pulse responses produced by the Have Derivatives process were achieved using a broadband elevator input. This is because the high frequency content of the broadband input best approximates the frequency content of an impulse. Consequently, the discrete pulse responses, which were shown to correlate with the simulated ideal impulse, also contained the high frequency content. Also, ensemble averaging is most effective when the input has uniformly distributed frequency content (Flying Qualities Branch, 1992:25-60 to 25-61). The broadband input function possesses uniformly random frequency content making it the optimal input to use with the Have Derivatives process as compared to the other input functions of Figure 4.10. For these reasons, all subsequent simulations that incorporated the Have

Derivatives process used the broadband input shown in Figure 4.10 and the results were compared against the baseline runs made with a doublet input.

Other than the choice of a particular window function, there are three parameters that can be varied to affect performance of the ensemble averaging step of the Have Derivatives process. Those parameters are the percent of section overlap, the section length, and the total maneuver length. A parametric analysis approach was taken to assess the impact that independently changing each of these parameters had on the accuracy of the stability derivative estimates. The assessment was made by comparing the rss magnitude of the simulation results from the Have Derivatives process with PCPID to the baseline doublet results from PCPID alone. The simulations incorporating the Have Derivatives process were made using the same conditions of rms turbulence intensity and measurement noise as the associated baseline simulations.

As mentioned in Chapter 2, a nonrectangular window can be applied to each section prior to ensemble averaging to reduce the amount of sidelobe leakage. A Hanning window was recommended, found to work well, and was therefore used for all Have Derivatives process simulations (Flying Qualities Branch, 1992:25-62; Rameriz, 1985:139 to 143).

4.3.1 Overlap Percent Simulation Results. The percentage of section overlap affects the number of sections that can be averaged for a given maneuver length according to Equation (2.3).

$$\text{Number of Sections} = \frac{\text{Total Signal Length} - \text{Overlap Length}}{\text{Section Length} - \text{Overlap Length}} \quad (2.3)$$

where overlap length is equal to the product of percentage of section overlap and section length.

In theory, the more sections that are averaged, the more the variance is reduced (Krauss and others, 1994:1-64 to 1-67). However, the amount of overlap has a practical limit as discussed in

Chapter 2. Fifty percent overlap has been shown to be very effective at minimizing signal variance (Krauss and others, 1994:1-66; Porat, 1994:112; Welch, 1967:72). Simulations were made using 0% and 50% overlap to investigate the effects of overlapping sections.

The maneuver length was set to 2048 points for these tests. Calm conditions of 3 ft/sec rms turbulence intensity and 10% measurement noise level were used for both the PCPID alone simulation and the Have Derivatives process with PCPID simulation. A sample length and window length of 1024 points was used for these runs. Figure 4.12 presents the results of the overlap percent simulations.

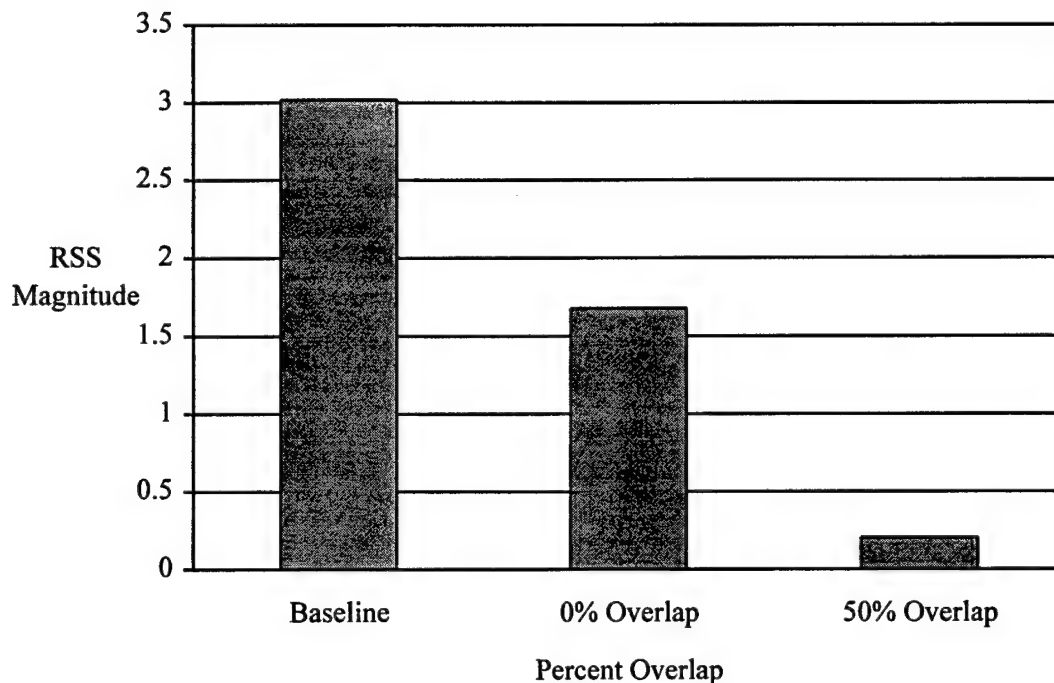


Figure 4.12. Root-Sum-of-Squares Magnitude versus Percent Overlap

Clearly, 50% overlap provided the best results and was used throughout the remainder of the simulations using the Have Derivatives process with PCPID.

4.3.2 Section Length Simulation Results. The next parameter varied was the section length. As with percent of overlap, section length affects the number of sections that can be averaged for a given maneuver length. A shorter section length results in more sections to be averaged, theoretically reducing the variance of the signal (Porat, 1994:126).

Section length also affects the resolution of the frequency response function. A longer section length gives better resolution. The tradeoff is that longer section lengths also give less sections to average. With these considerations in mind, section lengths of 512 and 1024 points were chosen for the simulations. The maneuver length for these runs was again set to 2048 points (~30 sec) and 50% overlap was used.

Simulations were accomplished under all three combined conditions of rms turbulence intensity and measurement noise level with the results given in Figure 4.13.

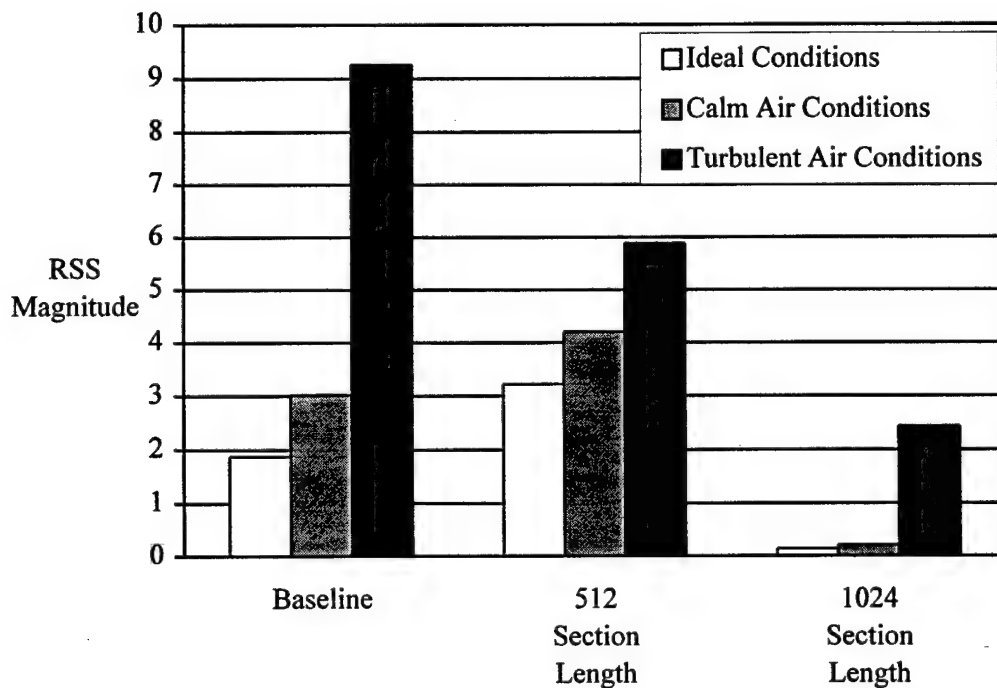


Figure 4.13. Root-Sum-of-Squares Magnitude versus Section Length

The results from the section length tests clearly indicate that 1024 points provided superior estimates to 512 points. This finding contradicted the expected outcome that 512 points would decrease the variance in the signals more than 1024 points, resulting in more accurate stability derivative estimates, and hence, lower rss magnitude. The explanation is found in the ensemble averaging algorithm.

Because the sections being averaged are not completely independent, there is some bias introduced with the averaging. In fact, there is a tradeoff between increasing bias and decreasing variance as the number of sections averaged for a given maneuver length are increased (Porat, 1994:126; Kay, 1988:73 to 74).

The recommended technique is to start with a small section length and make several runs with successively longer sections until there is no further improvement in estimation accuracy (Kay, 1988:73 to 74). This guidance explains the outcome and validates the selection of 1024 points as the best section length in this case.

4.3.3 Maneuver Length Simulation Results. The final parameter varied was maneuver length. For a given section length, the number of sections that can be averaged increases as maneuver length increases. By properly adjusting the combination of section length and maneuver length, smaller variances can be achieved without significantly increasing the bias in the averaged signal. The net result is better estimates with increasing maneuver length.

The maneuver length simulations were run at increasing maneuver lengths of 4096 and 8192 points. These approximately equated to one and two minutes, respectively, or double and quadruple the baseline maneuver length of 2048 points, or about 30 seconds, at a sampling rate of 67 sample per second. To isolate the effects of increasing maneuver length on estimate accuracy, the section length was kept at 1024 points with 50% overlap for both runs. Turbulent

conditions of 9 ft/sec rms turbulence intensity and 30% measurement noise were used for both simulations to provide a more challenging test of the averaging method.

The results of the maneuver length tests are shown in Figure 4.14.

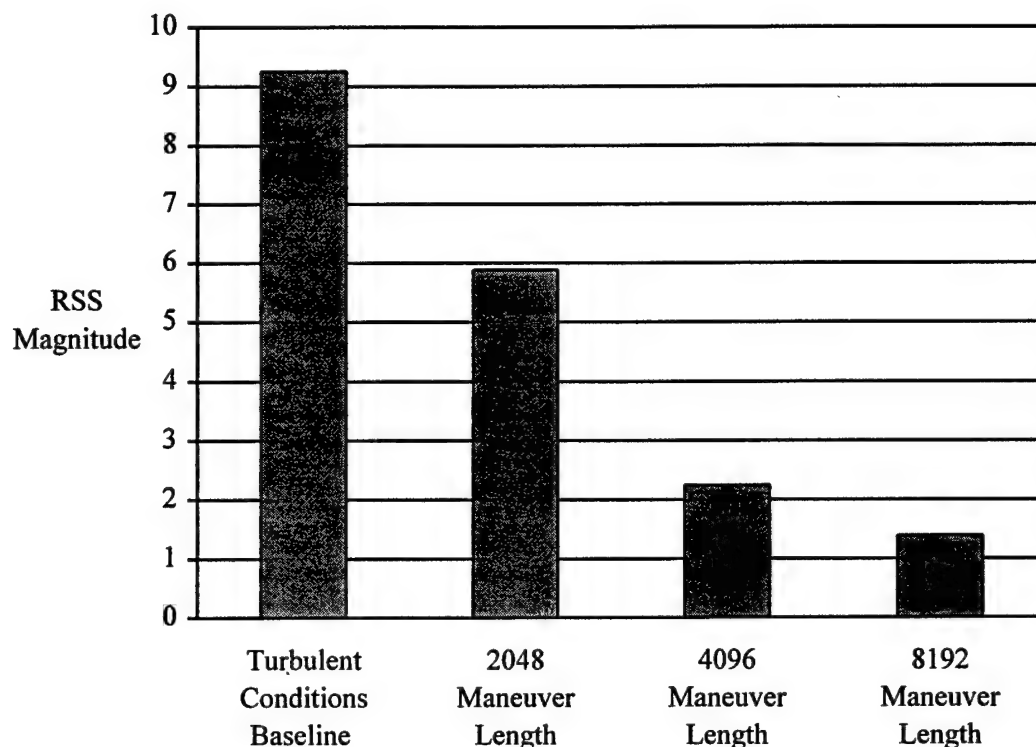


Figure 4.14. Root-Sum-of-Squares Magnitude versus Maneuver Length

As expected, increasing maneuver length did improve the stability derivative estimates. However, there is another tradeoff. Longer maneuver lengths cost more money, especially when talking about flight test time. Therefore, the prime consideration is to use the minimum maneuver length possible to achieve the required level of stability derivative estimate accuracy. A reasonable maneuver length for extracting stability derivatives related to the short period mode of most aircraft was determined to be around 30 seconds, or 2048 sample points at a sampling rate of 67 samples per second.

4.4 Summary

The following conclusions can be drawn from the simulation results using PCPID alone and using the Have Derivatives process with PCPID:

1. Increasing process and/or measurement noise levels resulted in less accurate stability derivative estimates.
2. The output error parameter estimation algorithm in PCPID performed best when given the impulse response.
3. The optimum amount of section overlap was found to be 50% for the given conditions of 1024 point section length and 2048 point maneuver length.
4. For a maneuver length of 2048 points and 50% section overlap, best results were achieved with a section length of 1024 points.
5. A maneuver length of 2048 points provided the best balance of accuracy and economy.

As a final illustration of the effectiveness of the Have Derivatives process to reduce signal variance, a run was made using a simulated ideal impulse elevator input and turbulent conditions. A plot of angle of attack versus time from this run is shown in Figure 4.15.

The same conditions were used for the simulation whose results are presented in Figure 4.16. The only difference was that a broadband signal was used as the input instead of a simulated impulse. The time history, however, became the discrete pulse response after using the Have Derivatives process. The marked decrease in signal variance was due to the use of ensemble averaging within the process.

Excellent results were achieved by the Have Derivatives process in simulation. The next logical step was to evaluate the process with flight test data. The next chapter outlines a flight test program to gather data necessary to satisfy the research objectives of Chapter 1.

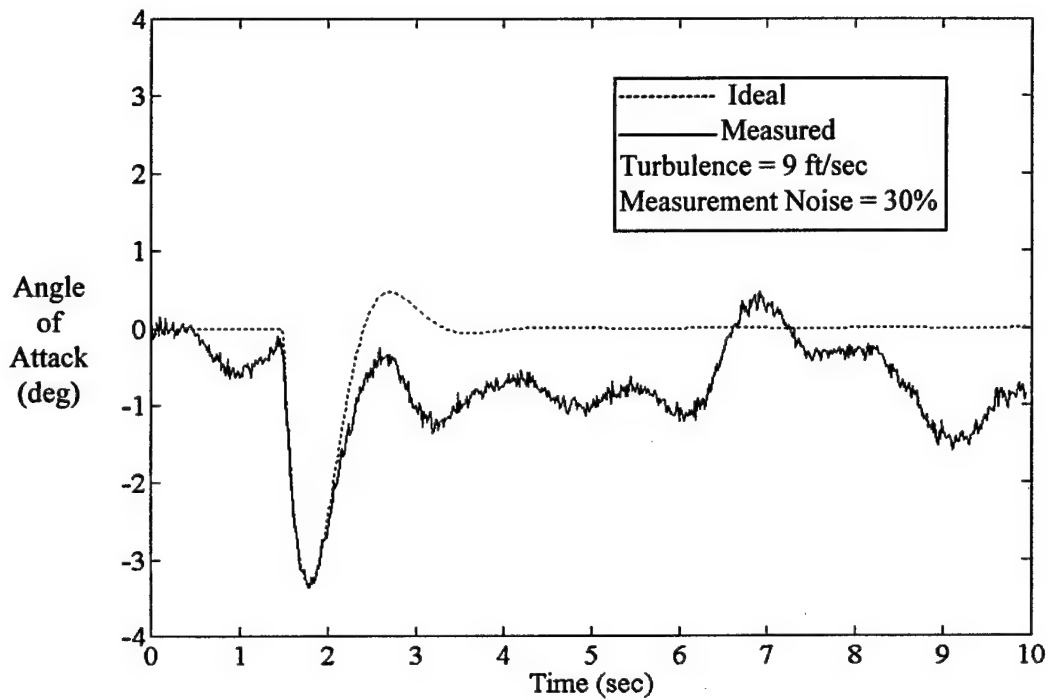


Figure 4.15. Turbulent Conditions, Impulse Response Without Have Derivatives Processing

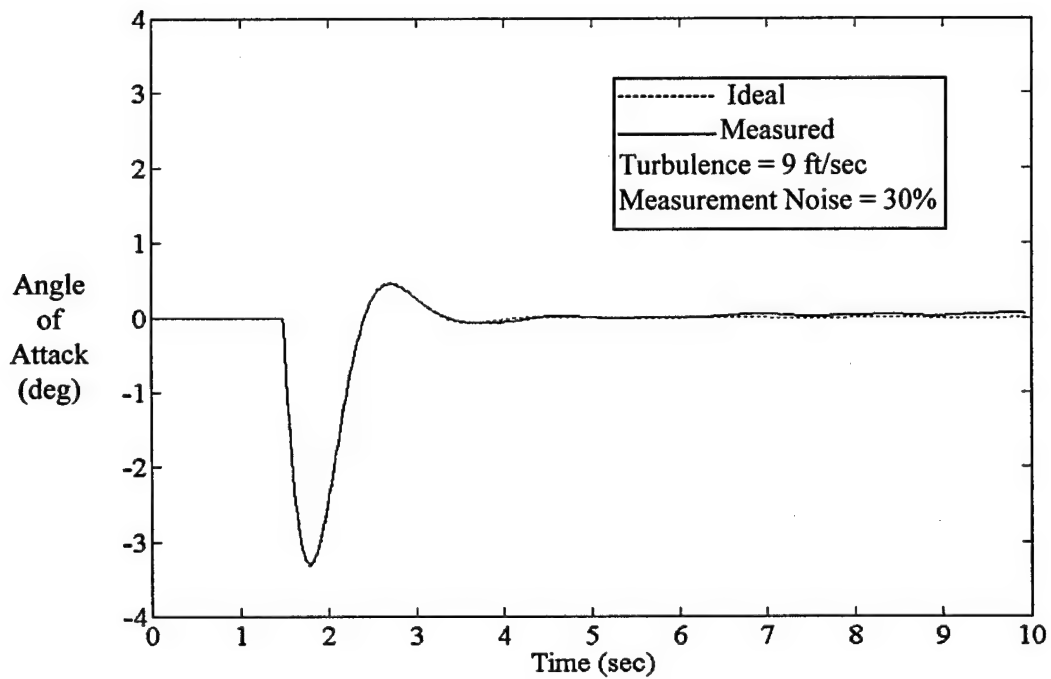


Figure 4.16. Turbulent Conditions, Discrete Pulse Response From Have Derivatives Process

5. Flight Test Development

5.1 Overview

A flight test program with the short title of Project HAVE DERIVATIVES was planned to evaluate the Have Derivatives process using actual flight data. More specifically, the plan was devised to evaluate the capability of the ensemble averaging step of the Have Derivatives process to reduce stability derivative estimation errors caused by turbulence.

The first main subsection of this chapter will be an introduction to the flight test plan, including a description of the F-16B test aircraft and definition of the main and specific flight test objectives. The planned flight test procedures are presented in the next main subsection followed by planned data reduction procedures in the last main subsection.

5.2 Introduction

The flight test program was planned for up to 12 sorties, at 1.2 hours each, in a modified Block 15 F-16B aircraft. The flight test assets, including the test aircraft, instrumentation and data reduction capabilities, were provided by the USAF Test Pilot School (TPS). All flights were planned to be launched from and recovered at Edwards AFB, CA and conducted within Air Force Flight Test Center (AFFTC) airspace using AFFTC facilities.

5.2.1 Test Aircraft Description. The planned test aircraft was a modified Block 15 F-16B. The F-16B was chosen because it had the best instrumentation fidelity of any TPS aircraft. Also, the TPS F-16B aircraft were used for parameter estimation demonstration rides in the TPS curriculum so there was already considerable experience and an adequate data base

established with the aircraft. Finally, the F-16B was used for a previous TPS flight test program aimed at evaluating in-house parameter estimation capability. The final report from that program was to be used for comparison for the Flight 1 validation step of Project HAVE DERIVATIVES. More detailed information about the standard Block 15 F-16B can be found in the F-16A/B Flight Manual, Technical Order 1F-16A-1 (Lockheed Fort Worth Company, 1997).

The planned test aircraft incorporated several modifications from the standard Block 15 F-16B configuration. External modifications were a yaw, angle of attack, pitot-static flight test noseboom mounted to the F-16B radome and an additional total temperature probe mounted at approximately the wing root, leading edge intersection. The external modifications had no impact on the flight test results.

Cockpit modifications were a sensitive airspeed indicator mounted in place of the production airspeed indicator and a modified altimeter, adjusted and calibrated for better accuracy, mounted in place of the production altimeter.

Finally, an airborne test instrumentation system (ATIS) data acquisition system (DAS) was installed on the aircraft. Although sampling rates of 100 to 200 samples per second are desired for parameter estimation, the ATIS DAS sampling rate of 67 samples per second was expected to be adequate (Maine and Iliff, 1986:105). A description of the ATIS DAS can be found in the USAF TPS Instrumentation Handbook (Instrumentation Branch, 1996). A complete description of all modifications to the test aircraft can be found in the F-16A/B Modification Flight Manual (Operations Branch, 1996).

5.2.2 Flight Test Objectives. An overall flight test objective and three specific flight test objectives were defined in accordance with the overall research objectives. The overall flight test objective was to perform a limited evaluation of the Have Derivatives process to reduce

stability derivative estimation errors caused by turbulence. The evaluation was limited due to time and resource constraints. Only errors due to process noise (turbulence) were to be investigated as the output-error parameter estimation algorithm within the Personal Computer Parameter Identification (PCPID) software did not account for process noise.

The overall flight test objective was further defined by the following three specific flight test objectives:

1. Establish a flight test basis data set of results against which to compare results of subsequent tests.
2. Compare calm air stability derivative estimation results using the Have Derivatives process with PCPID to the flight test basis data set results.
3. Compare turbulent air stability derivative estimation results using PCPID alone and using the Have Derivatives process with PCPID to the flight test basis data set results and to each other.

The next subsection outlines the flight test procedures planned to produce the flight data required to satisfy the flight test objectives.

5.3 Flight Test Procedures

Planned flight test procedures included the test aircraft configuration, flight test inputs and test point flight conditions.

5.3.1 Test Aircraft Configuration. Although there was not a requirement for a specific aircraft configuration, it was considered mandatory that the same configuration be used for the entire flight test program to ensure a fair comparison between data sets. The planned configuration was chosen to be similar to the aircraft configuration used in the previous

parameter estimation flight test program, AFFTC-TLR-92-12, F-16B Parameter Estimation (PEST) (Stambaugh and others, 1992:1). The planned configuration is presented in Table 5.1.

Table 5.1. Project HAVE DERIVATIVES
Test Aircraft Configuration

Component	Configuration
Landing Gear	RETRACTED
Leading Edge Flaps Switch	AUTO
Alternate Flaps Switch	NORM
Speedbrakes	CLOSED
Altimeter Setting	29.92 in Hg (PNEU)
Engine Air Source Knob	NORM
Air Conditioning Knob	AUTO
EEC/BUC Switch	EEC
Probe Heat Switch	ON
Anti-Ice Switch	AUTO
Loading Category Switch	I
Center of Gravity Range	35.88 - 37.38 pct MAC
Stations 1 and 9 (Wingtips)	16S210 Missile Rails
Station 5 (Centerline)	300 Gallon Fuel Tank

- Notes:
1. AUTO - automatic
 2. NORM - normal
 3. in Hg - inches of Mercury
 4. PNEU - pneumatic
 5. EEC - electronic engine control
 6. BUC - back up control
 7. pct MAC - percent of mean aerodynamic chord

The only deviation from the previous flight test program was the external centerline fuel tank that was planned to be carried for Project HAVE DERIVATIVES test flights. The tank was included to increase flight time and because that was the normal configuration for TPS F-16B's. If the original test aircraft could not be used for the full test program, the centerline tank configuration made it easier to transition to another TPS F-16B while maintaining the same

configuration. And because the external fuel tank was located along the aircraft centerline and near the aircraft center of gravity, the short period dynamics were not affected appreciably.

5.3.2 Flight Test Inputs. Each 1.2 hour test flight was planned to consist of approximately 15 data collection passes. A pass was defined as a 10-second trim shot, followed by a series of three pitch doublets, followed by a second 10-second trim shot, followed by a pitch frequency sweep of approximately 60 seconds in duration.

The trim shots were included to assess the level of turbulence prior to a particular data collection pass. Table 5.2 presents turbulence level definitions used during the HAVE DERIVATIVES flight test program.

Table 5.2. Project HAVE DERIVATIVES
Turbulence Level Definitions

Turbulence Level	Δ Normal Acceleration (Δg) ¹
Calm Air	$ \Delta n_z \leq 0.1$
Turbulent Air	$0.3 \leq \Delta n_z $

¹Change in normal acceleration from 1-g at a frequency of 1 radian per second

The turbulence level was defined based on the average normal acceleration deviation at the test aircraft's short period undamped natural frequency, predicted to be approximately 1 radian per second. Throughout this report, turbulence refers only to vertical disturbances, which were expected to have primary influence on the longitudinal stability derivative estimation results. The delta normal acceleration was to be measured at the test aircraft's estimated short period undamped natural frequency since it was reasoned that turbulence at that frequency would have

the greatest impact on the aircraft's longitudinal response, and consequently, the stability derivative estimation results.

The values in Table 5.1 were chosen based on simulation results and previous flight data. Normal acceleration typically varied only 0.05 g from 1 g trim flight in previous F-16B calm air data collected during early morning flights at low altitude. The maximum calm air normal acceleration deviation value of 0.1 g was chosen to allow for slight differences that might be observed during flight testing, but that would not have had an appreciable effect on the accuracy of stability derivative estimation results. The turbulent air value was chosen to ensure that there would be enough turbulence to see an effect on the aircraft responses, and consequently greater errors in the derivative estimates, while not being so large that the turbulence level might not be encountered during the test program.

Trim shots were planned to be approximately 10 seconds, hands-off, with the aircraft straight, level and unaccelerated, within trim shot data bands and tolerances defined in Table 5.3.

Table 5.3. Project HAVE DERIVATIVES
Trim Shot Data Bands and Tolerances

Parameter	Data Band ¹	Tolerance ²
Altitude (ft)	± 1000	± 100
Mach Number	± 0.02	± 0.01

¹Maximum acceptable deviation about the desired test parameter

²Maximum acceptable deviation about the initial trim shot data point value for the current trim shot

The requirements in Table 5.3 were chosen based on stability derivative sensitivity to variations in altitude and Mach number in previous F-16 flight test programs (Pape and Garland, 1979:181 to 195; Kelleher and Milam, 1982:225 to 238). The data band was the maximum

acceptable deviation about the desired test parameter while the tolerance was the maximum acceptable deviation about the initial trim shot data point value for the current trim shot.

Following the trim shot, the pilot was to accomplish three successive pitch doublets. The pilot was expected to perform a pitch doublet, wait for the aircraft response to subside, and then repeat a second and then a third pitch doublet. The control room, if available, would monitor angle of attack (AOA) and pitch rate to ensure the aircraft stayed within ± 5 degrees of trim AOA and within ± 20 degrees per second pitch rate. These limits were set to ensure that the responses did not violate linearity assumptions used in developing the Have Derivatives process. The control room, if available, would advise the pilot of maneuver quality following each set of three pitch doublets.

After the pitch doublets, the pilot was to accomplish a second straight, level and unaccelerated trim shot aiming for the same target flight condition for the same reasons as the first trim shot. The data bands and tolerances in Table 5.3 also applied for the second trim shot.

Following the second trim shot, the pilot was to perform a pitch frequency sweep lasting approximately 60 seconds. The sweep was planned to be initiated at a frequency well below the test aircraft predicted short period undamped natural frequency of approximately 1 radian per second, and progress slowly to a frequency well above 1 radian per second. To ensure consistency between passes, and between pilots, a system of counting during the inputs was devised. The first input would be increasing stick aft for 3 seconds followed by increasing stick forward for 6 seconds followed by increasing stick aft for 6 seconds. Because the first input was equal to one-quarter of the first cycle, the rest of the cycles were expected to be nearly symmetric about the trim flight condition. This would aid the pilot in keeping the aircraft about the trim flight condition during frequency sweeps, thereby minimizing chances that linearity assumptions would be violated. Subsequent frequency sweep cycles were to be decreased by one

count each cycle. The input was planned to be terminated after collection of 60 seconds of data over a frequency range from approximately 0.5 radians per second to 10 radians per second.

As with the doublets, the control room, if available, was to monitor angle of attack and pitch rate and inform the pilot if the limits of ± 5 degrees AOA or ± 20 degrees per second pitch rate were exceeded.

The parameters shown in Table 5.4 were to be tape recorded in-flight using the F-16B ATIS DAS. The same parameters were also to be recorded real-time to a computer file using the ATIS telemetry system installed aboard the test aircraft to provide a backup to the DAS. In addition, the flight test engineer in the rear cockpit of the test aircraft was expected to record true airspeed, Mach number, pressure altitude and total fuel quantity from rear cockpit gages during each trim shot.

Table 5.4. Project HAVE DERIVATIVES Parameters Recorded During Flight

Parameter Name	Type	Source	Range	Resolution	Samples per Sec
Time	Digital	Data Bus	0 to 99:59:59.99 hr:min:sec.milsec	1 milsec	66.67
Right Horizontal Stabilator Position	Analog	LVDT ^{1,2}	± 23.15 deg	0.2 deg	66.67
Angle of Attack	Analog	AOA Vane ^{3,4,5}	± 60 deg	0.04 deg	66.67
Pitch Angle	16 Bit Digital	INS	± 110 deg	0.005 deg	66.67
Pitch Rate	14 Bit Digital	HUD	± 45 deg/sec	0.022 deg/sec	66.67
Normal Acceleration	12 Bit Digital	HUD ^{6,7}	± 10 g	0.008 g	66.67
True Airspeed	15 Bit Digital	CADC	70 to 1700 KTAS (118.3 to 2873 fps)	0.125 KTAS (0.211 fps)	8.33
Mach Number	15 Bit Digital	CADC	0.1 to 3.0	0.0002	8.33
Pressure Altitude	16 Bit Digital	CADC	-1500 to 80,000 ft	2.5 ft	8.33
Total Fuel Quantity	Analog	Transducer	0 to 5100 lb	11 lb	8.33

Notes: 1. hr:min:sec.milsec - hour(s):minute(s):second(s).millisecond(s)

2. LVDT - Linear variable displacement transducer

3. deg - degree(s)

4. AOA - Angle of attack

5. INS - Inertial navigation system

6. HUD - Heads up display

7. CADC - Central air data computer

8. KTAS - knots true airspeed

9. fps - feet per second

10. ft - foot / feet

11. lb - pound(s)

¹Transducer was calibrated once before first test flight

²Five pole low-pass Butterworth filter with 9.5 Hertz cutoff frequency applied

³No corrections were made for upwash, boom bending or location from reference cg

⁴Angle of attack vane was calibrated once before first test flight

⁵Five pole low-pass Butterworth filter with 9.5 Hertz cutoff frequency applied

⁶Normal accelerometer was located at 6.87 feet aft of flight station 0.0, 0.64 feet left of centerline and 8.13 feet above ground

⁷No corrections were made for location from reference cg

Incompressible dynamic pressure, \bar{q} , was to be computed by the TPS Aydin processor post-flight using the following equation:

$$\bar{q} = 0.7 [2116.22 (1 - 6.87558 \times 10^{-6} H_c)^{5.2559}] M^2 \quad (5.1)$$

where H_c and M were measured in-flight.

5.3.3 Test Point Flight Conditions. Passes one through four of the first test flight were planned to be flown in calm air at 8,000 feet pressure altitude (PA) and 0.60 Mach number (M) to validate the test instrumentation and data reduction processes. The stability derivatives estimated from those four passes would be compared to stability derivatives previously estimated from data taken at the same flight condition, using a TPS F-16B and using the same flight test input. The previous results were presented in AFFTC-TLR-92-12, F-16B Parameter Estimation (PEST) (Stambaugh and others, 1992:8).

The remaining four passes from the first flight along with all other test flights were planned to be flown at 4,000 feet PA and 0.65 M. This test flight condition was chosen based on a tradeoff between the desire to fly as low to the ground as possible to encounter turbulence and the safety concerns associated with low altitude flight. Four thousand feet PA would provide at least 1,000 feet above ground level clearance along all routes expected to be flown during the test program. Also, this flight condition represented a 'heart of the envelope' point for the F-16B that would avoid the transonic region and allow data to be collected at an angle of attack that would result in a nearly linear aircraft response.

The first three flights were planned early in the day to gather calm air data. In contrast, Flights 4 through 12 were scheduled in the afternoon to gather data in turbulent air caused by winds and thermal convection.

The location of each test flight within AFFTC airspace was to be chosen by the aircrew based primarily on the level of turbulence desired for a particular flight. Every attempt would be made to remain sufficiently close to Edwards AFB to allow constant telemetry reception from the test aircraft. If a particular type of turbulence level, turbulent air or calm air, was initially encountered in a particular area, but part way through the test flight, that turbulence level no longer existed, the pilot was expected to maneuver to find the desired turbulence level.

Testing was planned to terminate when the test aircraft had the minimum amount of fuel necessary to return to base with required normal reserves.

5.4 Data Reduction Procedures

This subsection describes the steps planned to accomplish post-flight data processing. The first step in the data reduction process was to transfer flight data from the ATIS DAS tape to computer files using the TPS Aydin processor. Next, the TPS Flight Test Analysis Software (FTAS) could be used to convert the files from the Aydin processor's format to tab-delimited flat American Standard Code Information Interchange (ASCII) format (Data Processing Branch, 1996). The files could then be opened using MATLAB® software on a personal computer.

An interactive MATLAB® script file, entitled HDPROCES.M, was written by the Project HAVE DERIVATIVES team to automate the data reduction process. First the user would be asked to select the desired data pass file and ensure that its columns were correctly defined. Each data pass, consisting of the trim shots, the set of three pitch doublets and the pitch

frequency sweep was planned to be processed individually. Once the data pass file was loaded, the user would be prompted to define the bands to be used to identify erroneous data points that were clearly inaccurate in magnitude or sign. All data associated with the same time step as the erroneous data point would be removed, shortening the complete data file by one line each time that data were removed.

After removing bad data, HDPROCES would ask the user to identify a trim shot period on a plot of normal acceleration versus time. The program would calculate the power spectral density of normal acceleration for this period using the power spectral density routine (PSD.M) in the MATLAB® Signal Processing Toolbox® (Krauss and others, 1994:2-169 to 2-172). From the power spectral density, HDPROCES was written to calculate and return the value of normal acceleration deviation from 1 g at a frequency of 1 radian per second. The value of delta g given by HDPROCES was used to categorize the results of that particular pass based on the turbulence level definitions given in Table 5.2.

The user would then be prompted by HDPROCES to divide the data pass into two separate files, one containing the doublets and the other the pitch frequency sweep. HDPROCES would save the doublet input and response signals in GetData format as required by PCPID (Barnicki, 1996). The doublets were then ready to be processed by PCPID to estimate stability derivatives.

The pitch frequency sweep input and response signals would be further processed through the Have Derivatives process as described in Chapter 2. The Have Derivatives process would produce discrete pulse responses for angle of attack, pitch rate, normal acceleration and pitch angle. Those responses would be saved in the GetData format along with a simulated ideal impulse response that was independently created within HDPROCES. The simulated ideal impulse input and discrete pulse responses would then be ready for PCPID processing.

To achieve best results with PCPID it was important to update the aircraft mass and inertia model for each pass. The results presented throughout this report were for the cg locations specified for each pass and were not standardized to a particular cg location. The mass and inertia values for a particular pass were computed by a FORTRAN program called MASSCALC.EXE, written by Mr. Chris Nagy of Quartic Engineering Inc. (Nagy, 1997). Given the total fuel weight of the test aircraft it returned the mass, cg location, and moments and products of inertia. These values would then be hand recorded by the user and input into PCPID with the corresponding aircraft time histories.

No boom upwash or pitch rate effects corrections were made within HDPROCES to the angle of attack signal, however, corrections were made within PCPID for the location of the angle of attack and normal acceleration sensors relative to the aircraft cg for a given pass.

The complete code for HDPROCES.M and MASSCALC.EXE is presented in Appendix C, Data Reduction Programs.

After a data pass had been run through HDPROCES, parameter estimation would be accomplished using PCPID as described in Chapter 2. The output from PCPID would include stability derivative estimates, Cramer-Rao bounds that had been multiplied by a factor of 10 and matches of the PCPID computed response time histories to the measured response time histories.

Graphical output from HDPROCES would be copied and pasted into Microsoft® (MS) Word as a bitmap for report presentation. PCPID numerical results would be compiled in MS® Excel tables and figures and imported into MS® Word while the time history matches from PCPID would be saved as a bitmap file and imported into the report as well.

5.5 Summary

A flight test program with the short title of Project HAVE DERIVATIVES was planned to evaluate the Have Derivatives process using actual flight data. More specifically, the plan was devised to evaluate the ensemble averaging step of the Have Derivatives process to reduce stability derivative estimation errors caused by process noise, in the form of turbulence.

The test plan is summarized as follows:

1. The flight test program was planned for up to 12 sorties, at 1.2 hours each, in a modified Block 15 F-16B aircraft.
2. All flights were to be launched from and recovered at Edwards AFB, CA and conducted within Air Force Flight Test Center (AFFTC) airspace using AFFTC facilities.
3. An overall flight test objective and three specific flight test objectives were defined in accordance with the overall research objectives.
4. The planned aircraft configuration was chosen to be similar to the configuration used in a previous parameter estimation flight test program.
5. Each test flight was planned to consist of approximately 15 data collection passes with each pass composed of a 10-second trim shot, followed by a series of three pitch doublets, followed by a second 10-second trim shot, followed by a pitch frequency sweep of approximately 60 seconds in duration.
6. Passes one through four of the first test flight were to be flown in calm air at 8,000 feet pressure altitude (PA) and 0.60 Mach number (M) for the purpose of validating the test instrumentation and data reduction processes.
7. The remaining four passes from the first flight along with all other test flights were to be flown at 4,000 feet PA and 0.65 M.

8. The first three flights were planned early in the day to gather calm air data while Flights 4 through 12 were scheduled in the afternoon to gather data in turbulent air.
9. Flight data was to be tape recorded by an onboard DAS with backup recording via telemetry and by hand by the flight test engineer in the test aircraft rear cockpit.
10. Data would be transferred post-flight from the DAS tape to computer files which could then be processed on a personal computer using MATLAB® software.
11. An interactive, automated data processing routine called HDPROCES.M was written to remove wild points, calculate the turbulence level, run frequency sweep data through the Have Derivatives process, calculate aircraft mass, cg location, moments and products of inertia and save the pitch doublet and discrete pulse input and response signals in the correct format for PCPID input.
12. Output from HDPROCES was input for PCPID which would accomplish parameter estimation, returning stability derivative estimates, Cramer-Rao bounds multiplied by a factor of 10, and time history matches.
13. Graphical output from HDPROCES and PCPID would be presented in MS® Word as bitmap images for report presentation.
14. Numerical output from PCPID would be compiled in MS® Excel tables and figures which would be imported into MS® Word for reporting.

The flight test plan developed in this chapter was executed successfully by the Project HAVE DERIVATIVES team. Extensive data were gathered and reduced using the HDPROCES program described in this chapter. The flight test results and analysis produced by this effort are presented next in Chapter 6.

6. Flight Test Results and Analysis

6.1 Overview

A flight test program was executed in accordance with the Project HAVE DERIVATIVES flight test plan presented in the last chapter. The program consisted of 11 sorties, totaling 15.2 hours, in F-16B, S/N 80-0635. All sorties were flown at the Air Force Flight Test Center (AFFTC), Edwards AFB, California, between 15 September and 7 October 1997.

Flight test results presented in this chapter include:

1. A validation of data reduction processes and test aircraft instrumentation using calm air results from Personal Computer Parameter Identification (PCPID) alone
2. The establishment of a flight test basis data set using calm air results from PCPID alone
3. A comparison of calm air results from the Have Derivatives process with PCPID to the flight test basis data set
4. A comparison of turbulent air results from PCPID alone to the flight test basis data set
5. A comparison of turbulent air results from the Have Derivatives process with PCPID to the flight test basis data set and to turbulent air results from PCPID alone

These results are analyzed and appropriate conclusions are drawn.

6.2 Calm Air Results From PCPID Alone (Validation)

The first four passes of Flight 1 were accomplished to validate the data reduction processes and test aircraft instrumentation. Pitch doublets were performed, as described in Chapter 5, at

8,000 feet pressure altitude (PA) and 0.60 Mach number (M), in calm air. The stability derivative values estimated using PCPID alone were compared to estimates of the same stability derivatives as presented in AFFTC-TLR-92-12, F-16B Parameter Estimation (PEST) (Stambaugh and others, 1992). The stability derivative estimates presented in that report were calculated using data taken at the same flight condition, using the same flight test technique and using PCPID alone. The comparison is presented in Table 6.1.

Table 6.1. Validation of Data Reduction Processes and F-16B Test Aircraft Instrumentation

Stability Derivative	AFFTC-TLR-92-12 Mean Value ¹	Flight 1 Mean Value ²
$C_{N\alpha}$ (per deg)	0.0623	0.0662
$C_{m\alpha}$ (per deg)	-0.0005	0.0006
C_{mq} (per rad)	-1.6064	-4.5858
$C_{m\delta e}$ (per deg)	0.0117	-0.0122

- Notes: 1. Flight Condition: 8,000 ft PA, 0.60 M, Calm Air
2. AFFTC-TLR-92-12, F-16B Parameter Estimation (PEST) (Stambaugh and others, 1992:8)
3. cg range: 36.76 to 36.50 percent mean aerodynamic chord

¹Based on 5 samples

²Based on 4 samples

Although the mean value of $C_{m\alpha}$ estimated from the validation passes was of a different sign than the mean $C_{m\alpha}$ from AFFTC-TLR-91-12, the fact that $C_{m\alpha}$ was so close to zero in both cases indicated that this sign difference was not significant. The low positive value of $C_{m\alpha}$ for the validation passes did indicate that the unaugmented F-16B possessed nearly neutral static longitudinal stability at that flight condition, as expected.

The difference in C_{mq} was accepted when it was noted that C_{mq} showed high sensitivity to data processing in previous studies and had less effect on the aircraft short term response than the other three stability derivatives shown in Table 6.1 (Nagy, 1997:1).

The opposite sign for $C_{m\delta_c}$ was due to the fact that an opposite sign convention for horizontal stabilator deflection was used in AFFTC-TLR-91-12, while the sign convention used for this research was in accordance with the current USAF Test Pilot School (TPS) and National Aeronautics and Space Administration (NASA) sign convention.

A value for $C_{N\delta_c}$ was calculated for these test passes but was not presented as there was no value given in previous reports against which to compare. The validation of $C_{N\delta_c}$ was based on the good consistency of the values for that derivative for each of the four test passes as well as the small Cramer-Rao bound for the estimate of $C_{N\delta_c}$ for each of the four test passes.

Overall, the derivative values presented in Table 6.1 were sufficiently close to each other to validate the data reduction processes and test aircraft instrumentation.

6.3 Calm Air Results From PCPID Alone (Flight Test Basis Data Set)

A flight test basis data set of calm air stability derivative estimation results was developed using PCPID alone on aircraft pitch doublet input and response data. This was considered the 'truth source' against which later results were compared.

Results from Flights 1, 2, and 3 were used to develop this data set. Table 6.2 presents the mean stability derivative estimate values for the flight test basis data set and the associated confidence levels for the desired confidence intervals. Also presented in Table 6.2 are the actual confidence intervals based on a 95% confidence level. The equation used to calculate 95% confidence level intervals is given at Appendix D, Calm Air Stability Derivative Estimation Results. The mean stability derivative estimate values for the flight test basis data set were the result of averaging 32 separate estimates. Stability derivative estimates for each pass in calm air are presented in Table D.1 of Appendix D.

Table 6.2. F-16B Calm Air Mean Estimate Values and Confidence Measures From PCPID Alone

Stability Derivative	Mean Estimate Value	Desired Confidence Interval	Confidence Level ¹ (percent)	Actual Confidence Interval ²
$C_{N\alpha}$ (per deg)	0.0714	$\pm 2.5E-03$	100.0	$\pm 8.9E-04$
$C_{N\delta_e}$ (per deg)	0.0132	$\pm 2.0E-03$	100.0	$\pm 6.8E-04$
$C_{m\alpha}$ (per deg)	0.0008	$\pm 2.5E-04$	66.1	$\pm 5.3E-04$
C_{mq} (per rad)	-3.1380	$\pm 4.0E-01$	86.3	$\pm 5.3E-01$
$C_{m\delta_e}$ (per deg)	-0.0114	$\pm 4.0E-04$	70.6	$\pm 7.7E-04$

- Notes: 1. PCPID - Personal Computer Parameter Identification
2. Calculations based on 32 samples
3. Flight Condition: 4,000 ft PA, 0.65 M, Calm Air
4. cg range: 37.38 to 36.15 percent mean aerodynamic chord

¹Confidence level based on desired confidence interval

²Actual confidence interval based on 95 percent confidence level

The desired confidence intervals were chosen prior to flight test based on previous experience which indicated that stability derivative values that fluctuate less than the chosen confidence interval will not significantly affect the characteristics of the aircraft response (Nagy, 1976:70 to 71; Roskam, 1979:235 to 236; Flying Qualities Branch, 1994:302). Since the desired confidence intervals for $C_{m\alpha}$ and $C_{m\delta_e}$ were very small, it was difficult to meet them based on a 95 percent confidence level. The actual $C_{m\alpha}$ and $C_{m\delta_e}$ confidence intervals were still considered very tight. Performing more passes could have decreased the $C_{m\alpha}$ and $C_{m\delta_e}$ confidence intervals to their desired values.

The confidence interval for C_{mq} was not met due to the inherent difficulty in estimating C_{mq} from flight data. Despite that, the actual confidence interval for C_{mq} was considered acceptable.

Table 6.3 provides a quantitative description of the variation and confidence in the stability derivative estimates of the flight test basis data set. This table presents the average Cramer-Rao

bound for each of the five stability derivatives as well as the standard deviation for each derivative based on 32 estimates.

The Cramer-Rao bound was a measure of PCPID confidence in each stability derivative estimate. The smaller the Cramer-Rao bound, the more confidence in the estimate. The Cramer-Rao bound values presented throughout this report were the actual bounds multiplied by a factor of 10. This was based on common practice to more closely match the size of the bounds to the scatter of the estimates and to account for the fact that real noise is not white as assumed by estimation theory (Maine and Iliff, 1986:135 to 137). Cramer-Rao bounds for each flight test basis data set run are presented in Table D.1 of Appendix D, Calm Air Stability Derivative Estimation Results.

Table 6.3. F-16B Calm Air Standard Deviations and Mean Cramer-Rao Bounds From PCPID Alone

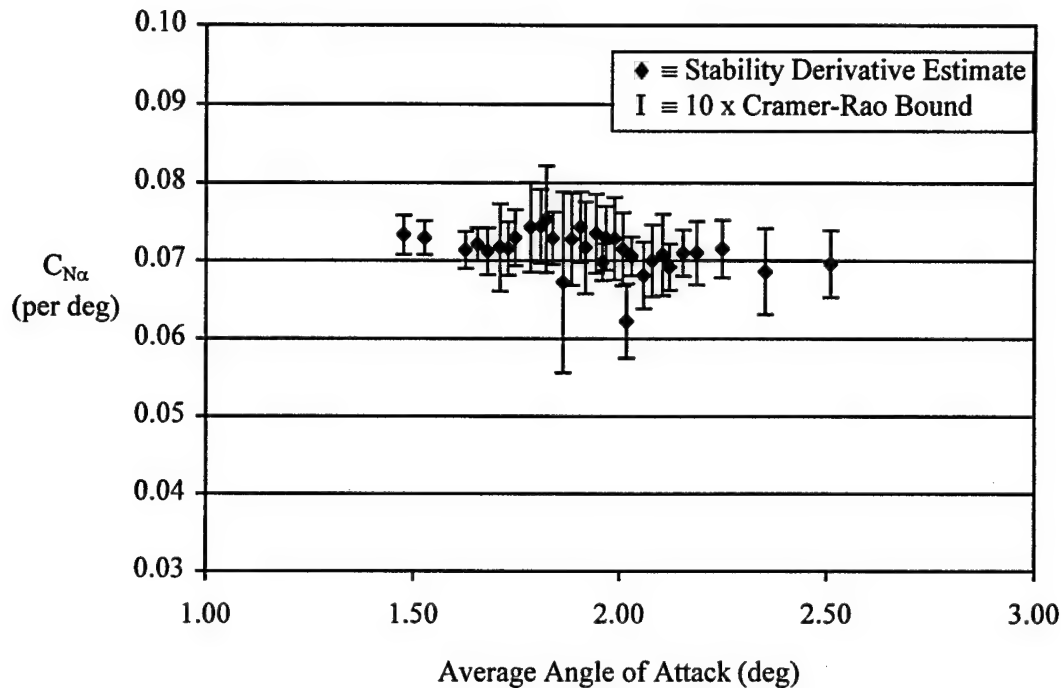
Stability Derivative	Standard Deviation	Mean Cramer-Rao Bound ¹
$C_{N\alpha}$ (per deg)	0.0025	$\pm 4.4\text{E-}03$
$C_{N\delta c}$ (per deg)	0.0019	$\pm 4.8\text{E-}03$
$C_{m\alpha}$ (per deg)	0.0015	$\pm 5.7\text{E-}04$
C_{mq} (per rad)	1.4808	$\pm 1.1\text{E-}00$
$C_{m\delta c}$ (per deg)	0.0021	$\pm 9.9\text{E-}04$

- Notes: 1. PCPID - Personal Computer Parameter Identification
 2. Calculations based on 32 samples
 3. Flight condition: 4,000 ft PA, 0.65 M, Calm Air
 4. cg range: 37.38 to 36.15 percent mean aerodynamic chord

¹Cramer-Rao bounds have been multiplied by a factor of 10

The small mean Cramer-Rao bounds in Table 6.2 indicated high PCPID confidence in the stability derivative estimates.

Figure 6.1 graphically depicts information similar Table 6.3. Instead of presenting the average Cramer-Rao bound for each stability derivative, as given in Table 6.3, Figure 6.1 presents the Cramer-Rao bound for each individual stability derivative estimate of $C_{N\alpha}$ in the basis data set. The estimate scatter can be seen as well in Figure 6.1.



- Notes: 1. Flight Condition: 4,000 ft PA, 0.65 M, Calm Air
 2. cg range: 37.38 to 36.15 percent mean aerodynamic chord
 3. Cramer-Rao bounds have been multiplied by a factor of 10
 4. PCPID - Personal Computer Parameter Identification

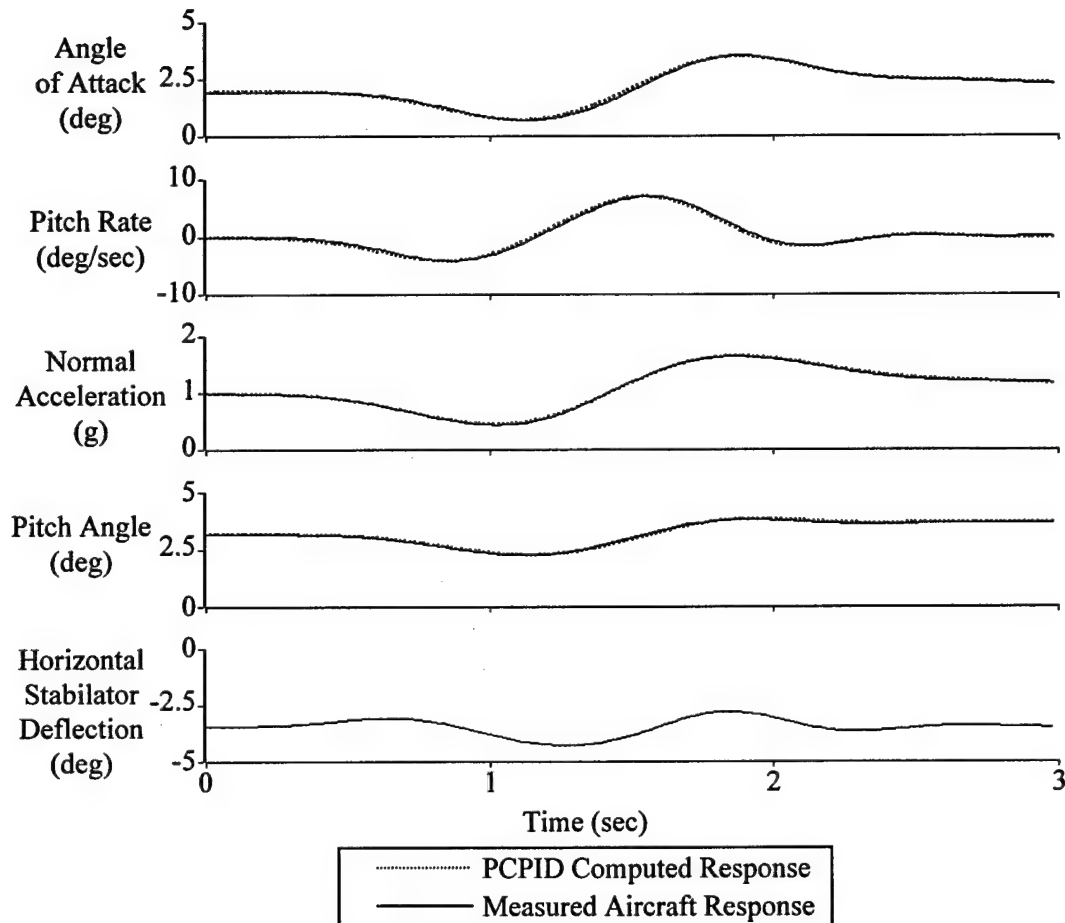
Figure 6.1. Example F-16B Calm Air Stability Derivative Estimates and Cramer-Rao Bounds From PCPID Alone

The scatter of points along the x-axis was due to varying angle of attack during flight as fuel was used and the center of gravity shifted.

Large scatter along the y-axis indicated less precision, and hence, less confidence in the estimate values. Associated with that scatter was the size of the Cramer-Rao bound as displayed

by bars for each estimate. In general, the scatter and Cramer-Rao bounds of the flight test basis data set were small, leading to high confidence in the precision and accuracy of the estimates.

Figure 6.2 presents a representative comparison between the test aircraft actual responses to the given doublet input and the PCPID computed time histories.



- Notes: 1. PCPID - Personal Computer Parameter Identification
 2. Flight Condition: 4,000 ft PA, 0.65 M, Calm Air
 3. cg location: 36.50 percent mean aerodynamic chord

Figure 6.2. Example F-16B Calm Air Time History Matches From PCPID Alone

The closer the computed time histories were to the actual responses, the higher the confidence in the final PCPID stability derivative estimates. The excellent time history matches

shown in Figure 6.2 were typical for the flight test basis data set and were qualitatively judged to indicate high confidence in the derivative estimates used to create the computed time histories.

Complementing the qualitative time history matches were the cost function values. The cost function values were a quantitative indication of how closely the PCPID computed time histories matched the measured time histories. The derivative estimates for a run with smaller cost function values were considered more accurate than those estimates associated with larger cost function values. The values of the cost functions were equalized at unity for the first 10 calm air passes by varying response parameter weightings within PCPID. These weightings were then used for all subsequent PCPID processing.

Table 6.4 presents the average cost function value for each response parameter used to develop the flight test basis data set along with the associated weightings used in PCPID.

Table 6.4. F-16B Response Weighting and Calm Air
Mean Cost Function Values From PCPID Alone

Response Parameter	Weighting Value	Mean Cost Function Value ¹
Angle of Attack (deg)	23	3.24
Pitch Rate (deg / sec)	3	2.11
Pitch Angle (deg)	14	1.85
Normal Acceleration (g)	60	3.48

- Notes: 1. PCPID - Personal Computer Parameter Identification
 2. Flight condition: 4,000 ft PA, 0.65 M, Calm Air
 3. cg range: 37.38 to 36.15 percent mean aerodynamic chord

¹Based on 32 samples

The small mean cost function values for the flight test basis data set correlated with the excellent time history matches shown in Figure 6.2.

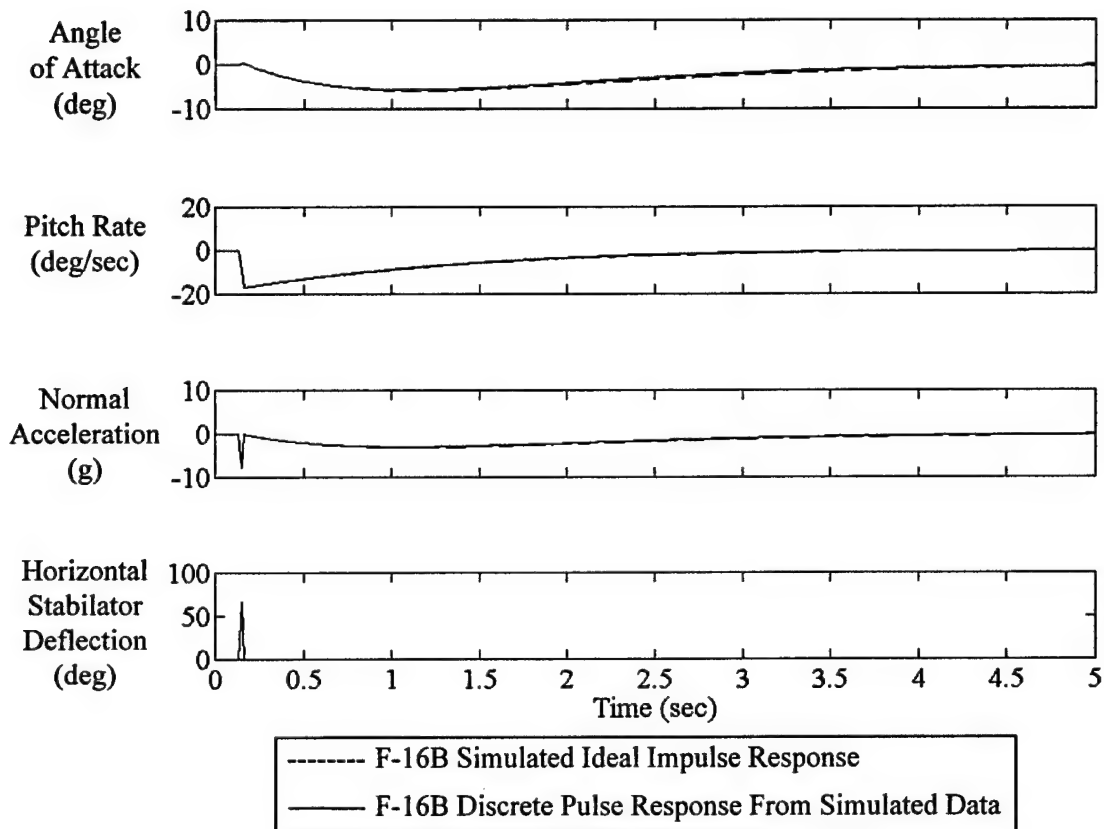
6.4 Calm Air Results From the Have Derivatives Process With PCPID

This section presents the calm air stability derivative estimation results found using the Have Derivatives process with PCPID. Results from simulations using an F-16B computer model are presented first to lay the foundation for the flight results that follow.

The HD_SIM program described in Chapters 3 and 4 was modified and used to validate the Have Derivatives process with an F-16B model prior to flight test. The flight condition in the program was changed to the desired Have Derivatives test condition of 4,000 feet PA and 0.65 M. Another modification included changing the aircraft model in the program to that of the F-16B test aircraft. The longitudinal stability derivatives were changed to the mean values from the flight test basis data set, as presented in Table 6.2, rounded to one significant digit. Finally, the mass and inertia properties of the program model were changed to reflect the test aircraft's weight and balance condition midway through a test mission, assuming standard fuel burn.

The first simulation resulted in divergent angle of attack, pitch rate and normal acceleration responses to a simulated ideal impulse input. This was due to the positive $C_{m\alpha}$ value of 0.0008 per degree. To produce a dynamically stable longitudinal response, $C_{m\alpha}$ was changed to -0.0001 per degree and the simulation was repeated. It should be noted that although the mean value of $C_{m\alpha}$ from the basis data set of 32 samples was 0.0008 per degree, there were 11 passes where $C_{m\alpha}$ was estimated to be equal to or more negative than -0.0001 per degree, with the most negative being -0.0012 per degree. The results of the second simulation, presented in Figure 6.3, were much better. Simulated ideal impulse responses are presented as dotted lines while the discrete pulse responses are solid lines. The simulated ideal impulse responses were generated by MATLAB's® LSIM.M routine as described in Chapter 3 using the stabilator simulated ideal

impulse input shown in Figure 6.3. The discrete pulse responses were produced from running a broadband input and responses through the Have Derivatives process.



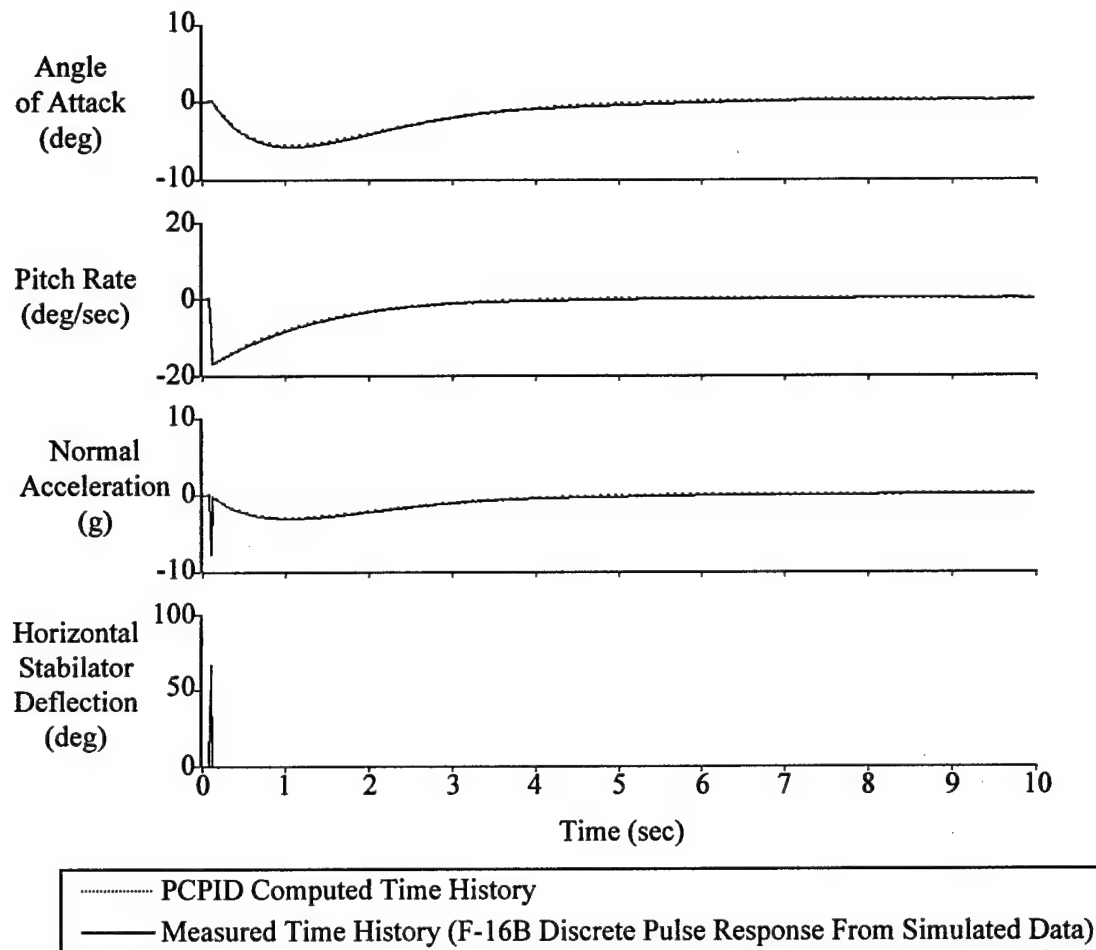
- Notes: 1. Flight Condition: 4,000 ft PA, 0.65 M
 2. cg location: 36.76 percent mean aerodynamic chord
 3. No process or measurement noise

Figure 6.3. F-16B Simulated Ideal Impulse Responses and Discrete Pulse Responses From Simulated Data

Figure 6.3 showed that the matches of the discrete pulse responses to the simulated ideal impulse responses were excellent, as with the AT-37B simulations.

The horizontal stabilator simulated ideal impulse input and the discrete pulse responses from the Have Derivatives process were entered into PCPID. The time history matches are presented

in Figure 6.4. The solid lines are the measured time histories while the dotted lines are the PCPID computed time histories.



- Notes:
1. PCPID - Personal Computer Parameter Identification
 2. Flight Condition: 4,000 ft PA, 0.65 M
 3. cg location: 36.76 percent mean aerodynamic chord
 4. No process or measurement noise

Figure 6.4. Matches of PCPID Computed Time Histories to F-16B Discrete Pulse Responses From Simulated Data

PCPID matched the simulated discrete pulse response time histories almost perfectly as with the AT-37B simulations. The numerical results are presented in Table 6.5 which contains the

actual derivative values used in the simulation model along with the PCPID estimates and their Cramer-Rao bounds.

Table 6.5. F-16B Noise Free Simulation, Actual and Estimated Derivative Values and Cramer-Rao Bounds From Have Derivatives Process With PCPID

Stability Derivative	Actual Value	Estimated Value	Cramer-Rao Bound ¹
$C_{N\alpha}$ (per deg)	0.0700	0.0714	$\pm 9.6E-04$
$C_{N\delta c}$ (per deg)	0.0100	0.0106	$\pm 6.9E-04$
$C_{m\alpha}$ (per deg)	-0.0001	-0.00005	$\pm 2.3E-05$
C_{mq} (per rad)	-3.0000	-3.3600	$\pm 1.3E-01$
$C_{m\delta c}$ (per deg)	-0.0100	-0.0102	$\pm 1.1E-04$

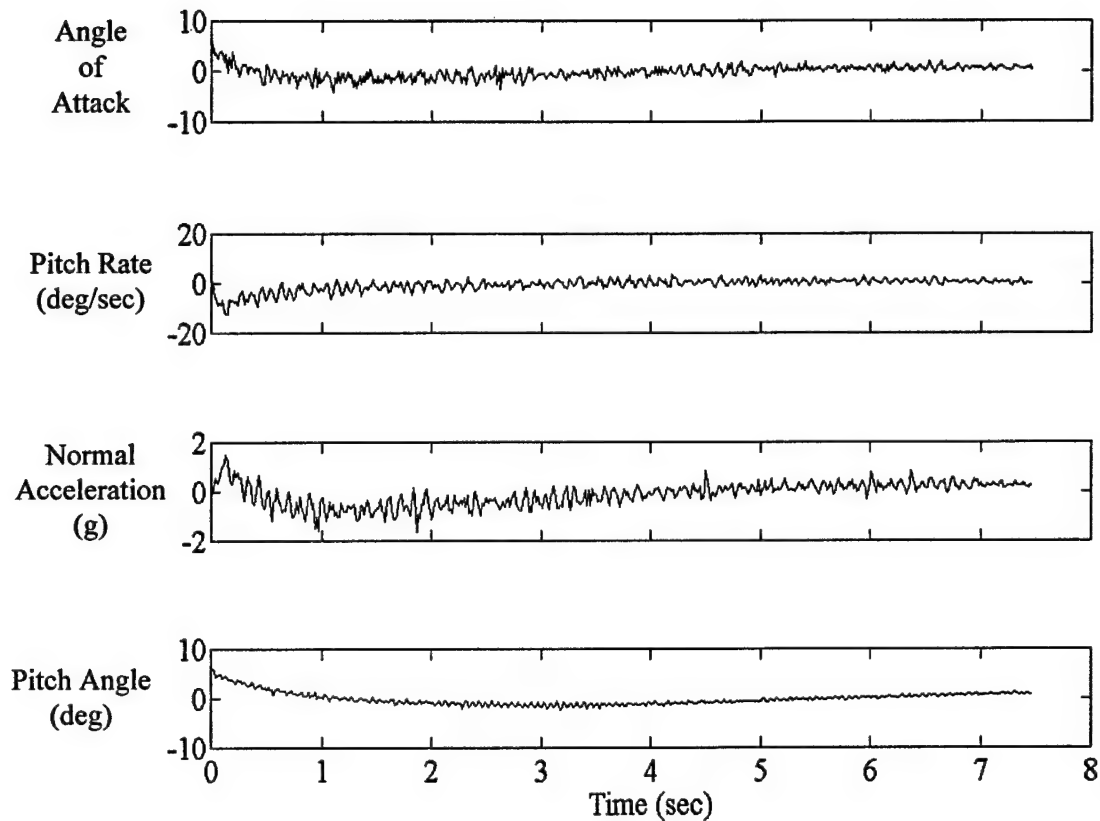
- Notes: 1. PCPID - Personal Computer Parameter Identification
2. Flight condition: 4,000 ft PA, 0.65 M
3. cg location: 36.76 percent mean aerodynamic chord
4. No process or measurement noise

¹Cramer-Rao bounds have been multiplied by a factor of 10

The very small Cramer-Rao bounds indicated the high confidence PCPID had in the accuracy of the F-16B simulation estimates which were very close to the actual estimate values used to build the simulation model. The minor deviations were due to estimation errors induced by the Have Derivatives process as well as the nature of the parameter estimation problem. The deviations were acceptable and the Have Derivatives process was validated with the F-16B model. The calm air flight data results from the Have Derivatives process with PCPID are next.

Pitch frequency sweep input and response time histories were processed post flight using the MATLAB® script file HDPROCES.M as discussed in Chapter 5. The coherence and frequency response of each longitudinal response to the frequency sweep input was qualitatively evaluated for data quality during processing. Most calm air frequency responses were excellent up to about 10 radians per second before deteriorating due to high frequency noise. Calm air data

coherence was generally above 0.8 up to 10 radians per second. A typical set of calm air discrete pulse responses, rendered by the Have Derivatives process, are shown in Figure 6.5.

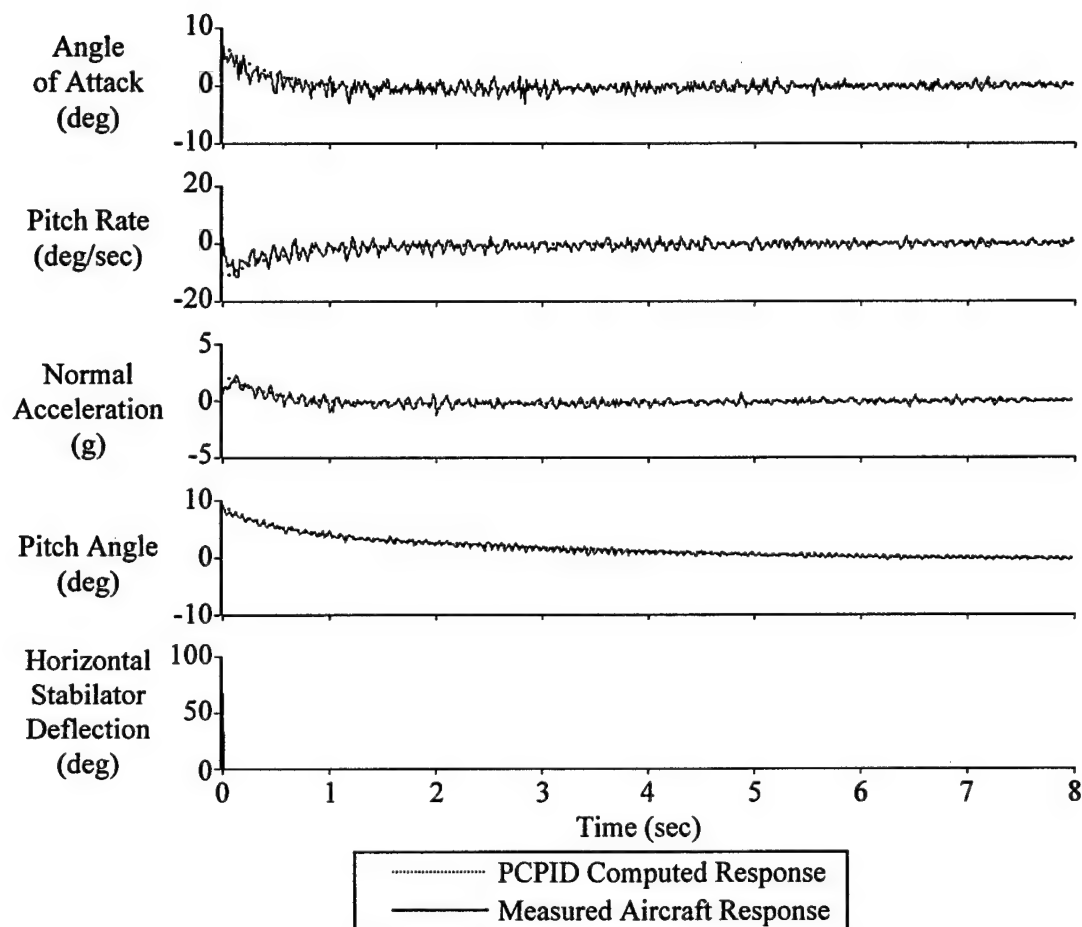


- Notes: 1. Flight Condition: 4,000 ft PA, 0.65 M, Calm Air
2. cg location: 37.03 percent mean aerodynamic chord

Figure 6.5. F-16B Calm Air Discrete Pulse Responses From the Have Derivatives Process

The high frequency noise noted in the average frequency responses was transformed into the time domain and degraded the discrete pulse responses shown in Figure 6.5. Although the responses were generally of the correct shape, the high frequency noise in the signals made it difficult for PCPID to estimate derivatives.

The responses shown in Figure 6.5 were imported into PCPID along with a simulated ideal impulse input. The resulting time history matches are shown in Figure 6.6.



- Notes: 1. PCPID - Personal Computer Parameter Identification
 2. Flight Condition: 4,000 ft PA, 0.65 M, Calm Air
 3. cg location: 37.03 percent mean aerodynamic chord

Figure 6.6. Matches of PCPID Computed Time Histories to F-16B Calm Air Discrete Pulse Responses From the Have Derivatives Process

Note that the simulated ideal impulse is located at the first time step to align with the first point of the discrete pulse responses. PCPID attempted to match the underlying responses by

disregarding the high frequency components and computing time histories that matched the general shape of the measured time histories. The resulting derivative values and Cramer-Rao bounds are given in Table 6.6, along with associated mean values from the basis data set.

Table 6.6. F-16B Calm Air Results From the Have Derivatives Process With PCPID and Associated Flight Test Basis Data Set Results

Stability Derivative	Basis Data Set Results Calm Air, PCPID Alone ¹		Calm Air Results Have Derivatives Process with PCPID ²		
	Mean Estimate	Mean Cramer-Rao Bound	Estimate	Cramer-Rao Bound	Deviation from Basis ³
$C_{N\alpha}$ (per deg)	0.0714	$\pm 4.4\text{E-}03$	0.0452	$\pm 1.1\text{E-}02$	10.48
$C_{N\delta c}$ (per deg)	0.0132	$\pm 4.8\text{E-}03$	0.0417	$\pm 6.6\text{E-}03$	15.00
$C_{m\alpha}$ (per deg)	0.0008	$\pm 5.7\text{E-}04$	0.0007	$\pm 4.2\text{E-}04$	0.07
C_{mq} (per rad)	-3.1380	$\pm 1.1\text{E-}00$	-6.4184	$\pm 1.2\text{E-}00$	2.22
$C_{m\delta c}$ (per deg)	-0.0114	$\pm 9.9\text{E-}04$	-0.0166	$\pm 2.0\text{E-}03$	2.48

- Note: 1. PCPID - Personal Computer Parameter Identification
2. Flight condition: 4,000 ft PA, 0.65 M, Calm Air
3. cg location: 37.03 percent mean aerodynamic chord
4. Cramer-Rao bounds have been multiplied by a factor of 10

¹Based on 32 samples

²Based on 1 sample

³Number of standard deviations from basis data set mean estimate

Although the estimate for $C_{m\alpha}$ appeared to be accurate, based on its small Cramer-Rao bound and small deviation from the basis data set mean estimate for $C_{m\alpha}$, the results for the other four derivatives were not acceptable. All four had a larger Cramer-Rao bound than the basis data set values for the same derivatives and each was at least two standard deviations from the basis data set mean estimate for their associated derivative. The results for $C_{N\alpha}$ and $C_{N\delta c}$ were particularly poor. Also, the Have Derivatives results were very sensitive to response weighting in PCPID, requiring considerable effort to achieve the results shown in Table 6.6.

The poor calm air Have Derivatives results led to the conclusion that the high frequency content on the discrete pulse responses had to be minimized if acceptable PCPID results were to be achieved.

One idea to minimize the noise was to append frequency sweep data from several passes to artificially increase the maneuver length, thereby increasing the number of sections that would be averaged within the Have Derivatives process. This approach required excessive flight data and produced only marginal improvement over processing one pass so it was not adopted.

The source of the problems appeared to be the lack of uniformly distributed frequency content in the frequency sweep flight test input. Recall that a broadband input was used in simulation with excellent results. Because the broadband input possessed nearly uniform frequency content, the ensemble averaging was effective at minimizing variance (noise). As more sections were averaged, the random components in the sections were minimized while the repetitive components, which defined the frequency responses of the aircraft, were amplified. In contrast, the frequency sweeps used in flight did not have uniformly distributed frequency content. The sweeps produced good frequency responses up to about 10 radians per second but were not effective with ensemble averaging. A uniformly distributed frequency content input could be applied through a programmed test input test set and should give better results with the ensemble averaging step of the Have Derivatives process.

Other approaches were investigated to minimize the noise when it was determined that the ensemble averaging was not effective enough with the frequency sweep flight test input. Because the contamination of the discrete pulse responses was due mainly to high frequency measurement noise, low-pass Butterworth filters were applied to the response signals prior to putting them through the Have Derivatives process.

A Butterworth type filter was chosen for its desirable property of being maximally flat in the pass-band and because it had been widely used for anti-aliasing in flight test instrumentation systems (Flying Qualities Branch, 1992:12-27).

Various combinations of filter order and frequency cutoff were examined. It was found that higher order filters could be used without noticeable increases in computation time. A ninth order filter was chosen as a good tradeoff between steep rolloff and filter complexity.

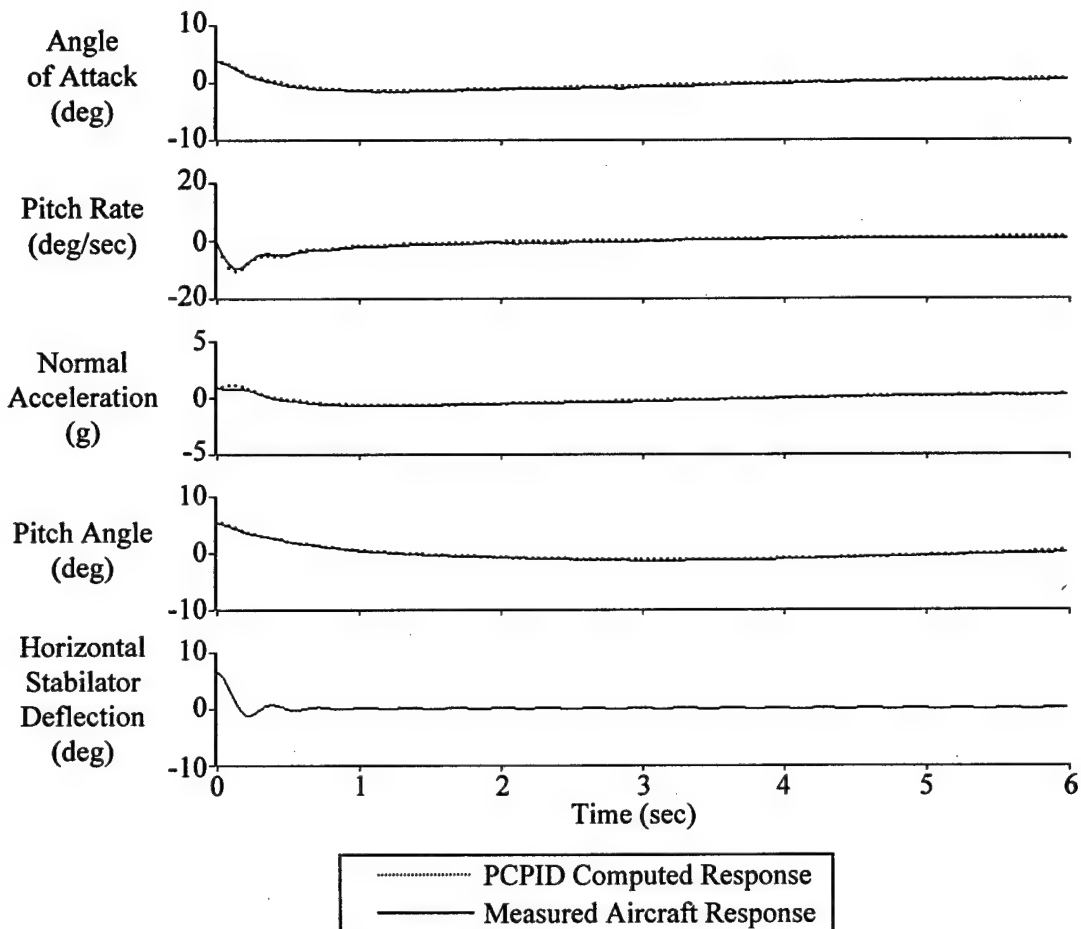
The cutoff frequency needed to be high enough so as not to distort the low frequency responses while being low enough to minimize the higher frequency noise. A cutoff frequency of 20 radians per second was determined to be the optimal choice for this research.

To avoid adding phase lag to the signals, the filters were applied both forward and reverse in time using the FILTFILT.M routine in MATLAB's® Signal Processing Toolbox®. Given all of the signal samples, the algorithm used information from the sample before and after each current sample to accomplish filtering with zero-phase distortion (Krauss and others, 1994:1-16). In addition to performing zero-phase digital filtering, the algorithm minimized startup transients by adjusting initial conditions to match the direct current component of each signal (Krauss and others, 1994:2-102).

The best results were achieved by filtering only the responses and not the frequency sweep input signal prior to Have Derivatives processing. However, there was concern that the simulated ideal impulse input would no longer correlate with the discrete pulse responses that had been shaped by the filters. That concern was validated with PCPID results when it was noted that the $C_{m\dot{\delta}_e}$ estimates were about half of the magnitude of the basis data set mean estimate of $C_{m\dot{\delta}_e}$.

To achieve a better match between the simulated ideal impulse input and the filtered discrete pulse responses, the same Butterworth filter was applied to the simulated ideal impulse signal

using the FILTFILT.M routine. The PCPID results from the filtered input and discrete pulse responses were noticeably better than without filtering. Typical time history matches are presented in Figure 6.7.



- Notes:
1. PCPID - Personal Computer Parameter Identification
 2. Flight Condition: 4,000 ft PA, 0.65 M, Calm Air
 3. cg location: 37.03 percent mean aerodynamic chord

Figure 6.7. Matches of PCPID Computed Time Histories to Filtered F-16B Calm Air Discrete Pulse Responses From the Have Derivatives Process

The time history matches presented in Figure 6.7 were very good with the only exception being the initial few points of the normal acceleration response. Figure 6.7 clearly shows that

the low-pass filters were able to eliminate most of the high frequency noise that was present on the same signals, shown unfiltered in Figure 6.6. Table 6.7 provides a comparison of the filtered Have Derivatives results along with the associated flight test basis data set results.

Table 6.7. F-16B Filtered Calm Air Results From the Have Derivatives Process With PCPID and Associated Flight Test Basis Data Set Results

Stability Derivative	Basis Data Set Results Calm Air, PCPID Alone ^{1,2}		Filtered Calm Air Results Have Derivatives Process with PCPID ^{3,4}		
	Mean Estimate	Mean Cramer-Rao Bound	Estimate	Cramer-Rao Bound	Deviation from Basis ⁵
$C_{N\alpha}$ (per deg)	0.0714	$\pm 4.4\text{E-}03$	0.0661	$\pm 4.2\text{E-}03$	2.12
$C_{N\delta e}$ (per deg)	0.0132	$\pm 4.8\text{E-}03$	0.0182	$\pm 5.1\text{E-}03$	2.63
$C_{m\alpha}$ (per deg)	0.0008	$\pm 5.7\text{E-}04$	0.0007	$\pm 1.1\text{E-}04$	0.07
C_{mq} (per rad)	-3.1380	$\pm 1.1\text{E-}00$	-7.0223	$\pm 3.9\text{E-}01$	2.62
$C_{m\delta e}$ (per deg)	-0.0114	$\pm 9.9\text{E-}04$	-0.0132	$\pm 5.3\text{E-}04$	0.86

- Notes: 1. PCPID - Personal Computer Parameter Identification
2. Flight condition: 4,000 ft PA, 0.65 M, Calm Air
3. Cramer-Rao bounds have been multiplied by a factor of 10

¹cg range: 37.38 to 36.15 percent mean aerodynamic chord

²Based on 32 samples

³cg location: 37.03 percent mean aerodynamic chord

⁴Based on 1 sample

⁵Number of standard deviations from basis data set mean estimate

Comparing the filtered Have Derivatives results from Table 6.7 to the unfiltered Have Derivatives results from Table 6.6 revealed that improvement was attained through the use of the filters. The Cramer-Rao bounds for all five filtered estimates were smaller than the unfiltered estimate Cramer-Rao bounds. In fact, the filtered estimate Cramer-Rao bounds were smaller than the flight test basis data set mean Cramer-Rao bounds for all derivatives but $C_{N\delta e}$, which was only slightly larger than the associated flight test basis data set value. This was considered significant as the comparison was made between the flight test basis data set mean Cramer-Rao

bounds, which were an average of 32 samples, and the single filtered Have Derivatives sample. The intermediate conclusion was that filtering resulted in smaller Cramer-Rao bounds which indicated greater confidence in the accuracy of the estimates.

The comparison of the filtered estimate deviations from the flight test basis data set to the unfiltered estimate deviations from the flight test basis data set was not as conclusive. Although the deviations of the filtered estimates were much smaller for C_{Na} , $C_{N\delta c}$ and $C_{m\delta c}$, the deviation of C_{ma} was the same while the deviation of the filtered C_{mq} was slightly larger. Also, three of the five filtered Have Derivatives estimates were still greater than two standard deviations from the basis data set mean estimates. That much deviation was considered unacceptable.

The filtering improved the Cramer-Rao bounds but the large deviations indicated that there were still some problems with the approach of using low-pass Butterworth filters to remove the noise. Although the FILTFILT.M routine had the desirable ability to filter the signals without adding phase lag, it did alter the amplitude of the signals slightly by a factor of the filter magnitude squared (Krauss and others, 1994:1-17). The amount of attenuation varied depending on the frequency content of the original signals. Those signals that had frequency content in the pass-band were less affected than those that contained frequencies above the pass-band. For example, the filters affected the high frequency simulated ideal impulse signal more than the low frequency responses, resulting in degraded correlation between the input and response signals.

Based on these results, for signals used in parameter estimation, it is recommended that low-pass filters not be used to remove noise unless absolutely necessary, and then only at a cutoff frequency high enough above the dynamics of interest so as not to corrupt those dynamics.

6.5 Turbulent Air Results From PCPID Alone

A flight test data set of turbulent air stability derivative estimation results was developed using PCPID alone on aircraft pitch doublet input and response data. The data were from those passes of Flights 4 through 11 where the normal acceleration deviations at the approximate short period undamped natural frequency of 1 radian per second were measured to be greater than 0.3 g. In this section, the turbulent air data set results are compared to associated flight test basis data set results, developed from calm air data using PCPID alone.

Table 6.8 presents mean stability derivative estimate values and confidence intervals based on a 95% confidence level. The equation used to calculate 95% confidence level intervals is given at Appendix E, Turbulent Air Stability Derivative Estimation Results. The stability derivative estimates for each pass in turbulent air are presented in Table E.1 of Appendix E.

Table 6.8. F-16B Turbulent Air Mean Stability Derivative Estimate Values and Confidence Intervals From PCPID Alone

Stability Derivative	Basis Data Set Calm Air, PCPID Alone ^{1,2}		Turbulent Air PCPID Alone ^{4,5}	
	Mean Estimate Value	Confidence Interval ³	Mean Estimate Value	Confidence Interval ⁶
$C_{N\alpha}$ (per deg)	0.0714	$\pm 8.9\text{E-}04$	0.0693	$\pm 1.67\text{E-}03$
$C_{N\delta_e}$ (per deg)	0.0132	$\pm 6.8\text{E-}04$	0.0174	$\pm 2.26\text{E-}03$
$C_{m\alpha}$ (per deg)	0.0008	$\pm 5.3\text{E-}04$	0.0010	$\pm 3.99\text{E-}04$
$C_{m\dot{\alpha}}$ (per rad)	-3.1380	$\pm 5.3\text{E-}01$	-3.5564	$\pm 7.91\text{E-}01$
$C_{m\delta_e}$ (per deg)	-0.0114	$\pm 7.7\text{E-}04$	-0.0111	$\pm 7.80\text{E-}04$

Notes: 1. PCPID - Personal Computer Parameter Identification
2. Flight Condition: 4,000 ft PA, 0.65 M

¹cg range: 37.38 to 36.15 percent mean aerodynamic chord

²Based on 32 samples

³Confidence interval based on 95 percent confidence level

⁴cg range: 37.38 to 35.88 percent mean aerodynamic chord

⁵Based on 52 samples

⁶Confidence interval based on 95 percent confidence level

The confidence intervals presented in Table 6.8 overlapped for all derivatives except $C_{N\delta e}$. Despite this, the turbulent air $C_{N\delta e}$ value was close enough to say the turbulent air derivative estimates calculated using PCPID alone were of the same population as the basis data set.

The standard deviation and mean Cramer-Rao bound are given in Table 6.9 for each derivative of the flight test basis data set and turbulent air data set. The Cramer-Rao bounds for each turbulent air data set run are presented in Table E.1 of Appendix E, Turbulent Air Stability Derivative Estimation Results.

Table 6.9. F-16B Turbulent Air Stability Derivative Standard Deviations and Mean Cramer-Rao Bounds From PCPID Alone

Stability Derivative	Basis Data Set Calm Air, PCPID Alone ^{1,2}		Turbulent Air PCPID Alone ^{3,4}	
	Standard Deviation	Mean Cramer-Rao Bound	Standard Deviation	Mean Cramer-Rao Bound
$C_{N\alpha}$ (per deg)	0.0025	$\pm 4.4\text{E-}03$	0.0060	$\pm 6.5\text{E-}03$
$C_{N\delta e}$ (per deg)	0.0019	$\pm 4.8\text{E-}03$	0.0081	$\pm 8.6\text{E-}03$
$C_{m\alpha}$ (per deg)	0.0015	$\pm 5.7\text{E-}04$	0.0014	$\pm 8.0\text{E-}04$
C_{mq} (per rad)	1.4808	$\pm 1.1\text{E-}00$	2.8403	$\pm 1.9\text{E-}00$
$C_{m\delta e}$ (per deg)	0.0021	$\pm 9.9\text{E-}04$	0.0028	$\pm 1.6\text{E-}03$

Notes: 1. PCPID - Personal Computer Parameter Identification

2. Flight Condition: 4,000 ft PA, 0.65 M

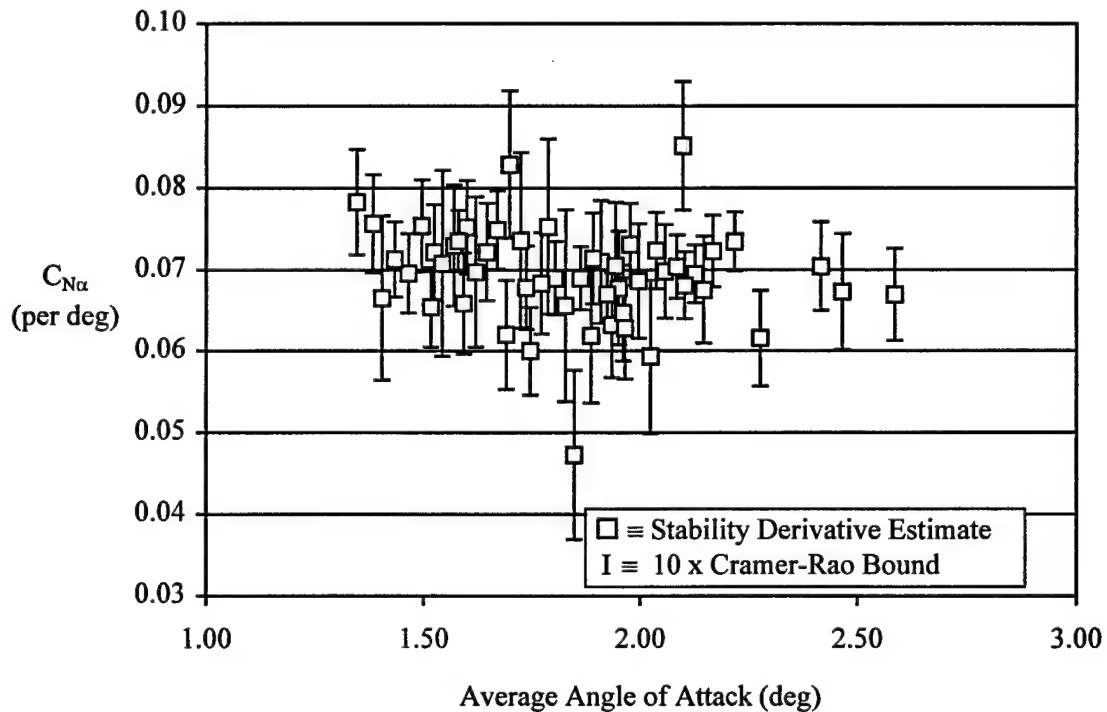
¹cg range: 37.38 to 36.15 percent mean aerodynamic chord ²Based on 32 samples

³cg range: 37.38 to 35.88 percent mean aerodynamic chord ⁴Based on 52 samples

The standard deviation for each turbulent air data set stability derivative was larger than its corresponding basis data set derivative except for $C_{m\alpha}$. However, the standard deviation of $C_{m\alpha}$ was the smallest for both data sets. Additionally, the turbulent $C_{m\alpha}$ estimate was nearly identical to the basis $C_{m\alpha}$ estimate and $C_{m\alpha}$ had the lowest mean Cramer-Rao bound in both sets. These

results indicated that PCPID was able to achieve its best results when estimating $C_{m\alpha}$, even when turbulence was present.

As expected, the turbulent data set mean Cramer-Rao bound was larger than the corresponding flight test basis data set mean Cramer-Rao bound for each of the derivatives. Overall, the standard deviation and mean Cramer-Rao bound results indicated that there was greater scatter within the turbulent data set and less confidence in the individual estimates. Figure 6.8 presents the Cramer-Rao bound for each individual estimate of $C_{N\alpha}$ in the turbulent air data set and shows the scatter of the turbulent data set estimates.

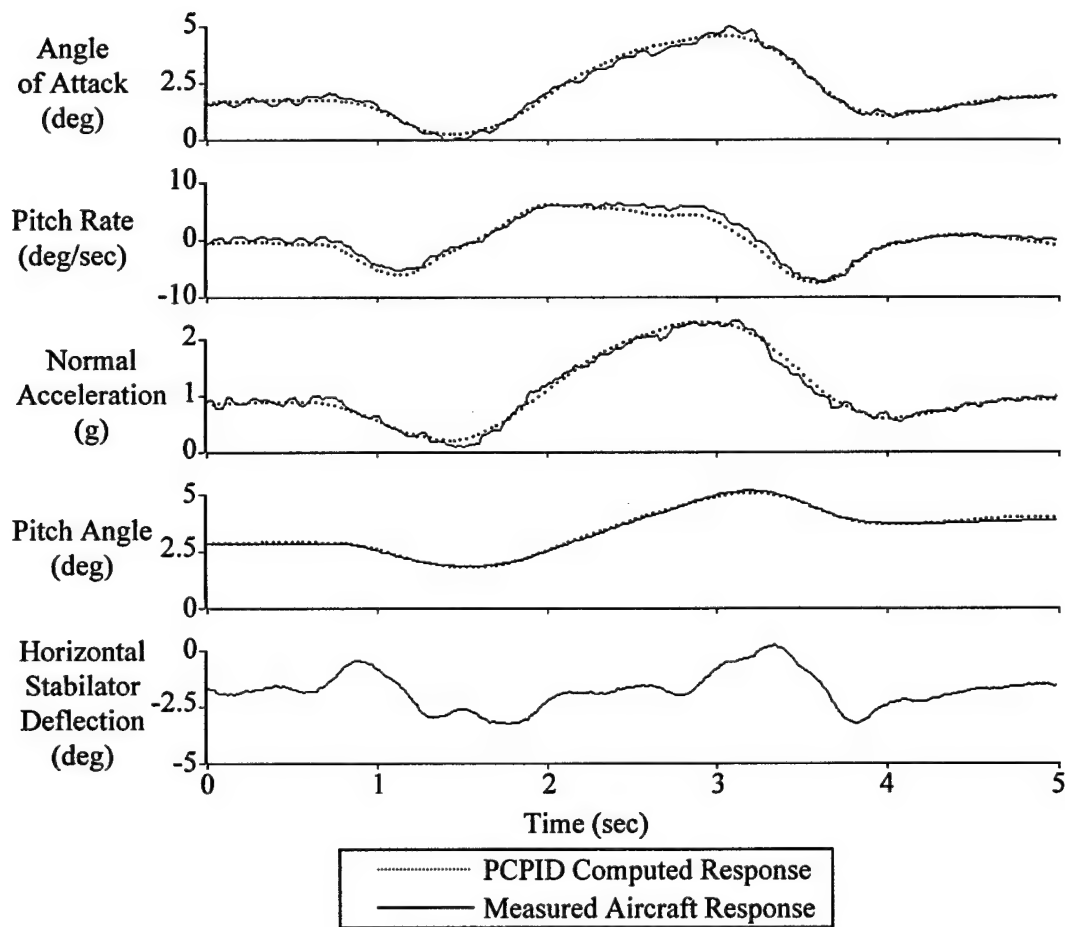


- Notes: 1. Flight Condition: 4,000 ft PA, 0.65 M, Turbulent Air
 2. cg range: 37.38 to 36.15 percent mean aerodynamic chord
 3. Cramer-Rao bounds have been multiplied by a factor of 10
 4. PCPID - Personal Computer Parameter Identification

Figure 6.8. Example F-16B Turbulent Air Stability Derivative Estimates and Cramer-Rao Bounds From PCPID Alone

The scatter of the turbulent air data along the vertical axis in Figure 6.8 was greater than the calm air (basis) data shown in Figure 6.1 and the individual turbulent air estimate Cramer-Rao bounds were generally larger. The larger vertical scatter in Figure 6.8 indicated less precision, and hence, less confidence in the estimate values, while the larger Cramer-Rao bounds for the turbulent air data set indicated lower confidence in the accuracy of the estimates.

Figure 6.9 presents a representative comparison between the test aircraft turbulent air time history responses to the given doublet input and the PCPID computed time histories.



- Notes: 1. PCPID - Personal Computer Parameter Identification
 2. Flight Condition: 4,000 ft PA, 0.65 M, Turbulent Air
 3. cg location: 36.85 percent mean aerodynamic chord

Figure 6.9. Example F-16B Turbulent Air Time History Matches From PCPID Alone

The turbulent air time history matches of Figure 6.9 were poor as compared to the calm air time history matches shown in Figure 6.2. The parameter estimation routine in PCPID had a difficult time discerning the true aircraft responses due to the doublet input from the response variations due to turbulence. The solution it converged to was a compromise that produced the poorer results discussed earlier in this section.

Complementing the qualitative time history matches were the numerical cost function values. Recall that the derivative estimates for a run with smaller cost function values were considered more accurate than those estimates associated with larger cost function values. Mean cost function values for the basis data set and turbulent air data set are shown in Table 6.10.

Table 6.10. F-16B Mean Cost Function Values From PCPID Alone

Output Response Parameter	Calm Air, PCPID Alone (Basis Data Set) ¹		Turbulent Air, PCPID Alone ³	
	Weighting Value	Mean Cost Function Value ²	Weighting Value	Mean Cost Function Value ⁴
Angle of Attack (deg)	23	3.24	23	16.88
Pitch Rate (deg / sec)	3	2.11	3	4.49
Pitch Angle (deg)	14	1.85	14	5.82
Normal Acceleration (g)	60	3.48	60	21.90

Notes: 1. PCPID - Personal Computer Parameter Identification
2. Flight Condition: 4,000 ft PA, 0.65 M

¹cg range: 37.38 to 36.15 percent mean aerodynamic chord

²Based on 32 samples

³cg range: 37.38 to 35.88 percent mean aerodynamic chord

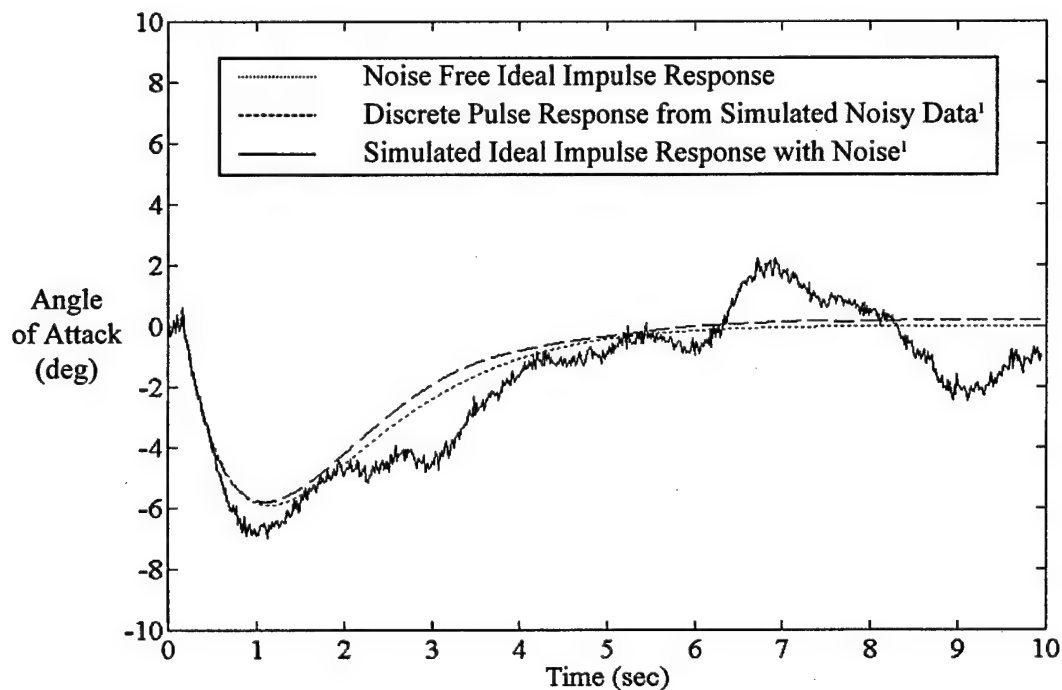
⁴Based on 52 samples

As expected, the turbulent air data set had larger mean cost function values which indicated less accuracy in the estimates given by PCPID for that data set as compared to the flight test basis data set. The higher cost function values for the turbulent air data set were due to poor time history matches like those shown in Figure 6.9.

6.6 Turbulent Air Results From the Have Derivatives Process With PCPID

Turbulent air results from the Have Derivatives process with PCPID are presented in this section. F-16B simulation results are first discussed to illustrate the effectiveness of the Have Derivatives process with the F-16B model. Problems encountered when using the Have Derivatives process with turbulent flight data are next presented including results obtained with the addition of low-pass Butterworth filters.

Simulated F-16B angle of attack responses are shown in Figure 6.10 to highlight the degradation due to process and measurement noises and to demonstrate the effectiveness of the Have Derivatives process at minimizing noise induced variance through ensemble averaging.



- Notes: 1. Flight Condition: 4,000 ft PA, 0.65 M
2. cg location: 36.76 percent mean aerodynamic chord

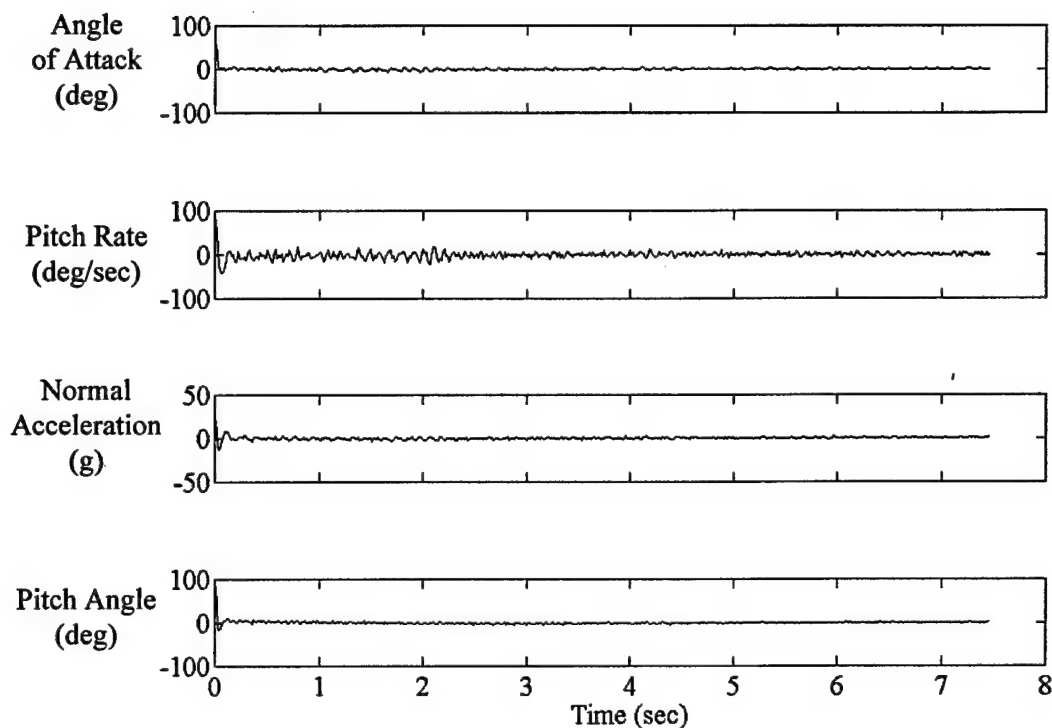
¹Same noise used for both cases

Figure 6.10. F-16B Noise Free and Noisy Simulated Angle of Attack Responses

The simulation was accomplished with the F16_SIM program discussed earlier. The noise free ideal impulse response was included as a reference. The same level of process and measurement noise was used for both the discrete pulse response and simulated ideal impulse response. The discrete pulse response was created by processing a broadband input and response through the Have Derivatives process. The simulated ideal impulse response was produced by a linear simulation with the simulated ideal impulse as the input. The simulated ideal impulse response represented a typical turbulent air signal that could be given to PCPID to estimate stability derivatives. The impact of that level of turbulence was covered in the previous section. The discrete pulse response demonstrated the effectiveness of the Have Derivatives process. The noise was minimized by the process to give a response that nearly matched the noise free ideal impulse response.

The angle of attack, pitch rate and normal acceleration discrete pulse responses from simulated noisy data were imported into PCPID along with a simulated ideal impulse input as before. PCPID was able to match the responses well and rendered derivative results that were as good as the calm air simulation results presented in Table 6.5. Next is a description of the difficulties encountered when attempting to use the Have Derivatives process on flight test data.

Unfortunately, the problems that were encountered with the calm air flight data using the Have Derivatives process were again encountered with turbulent air flight data. Lack of uniformly distributed frequency content in the frequency sweep flight test input and higher levels of turbulence resulted in unreasonable discrete pulse responses from the Have Derivatives process. A typical set of turbulent air discrete pulse responses are shown in Figure 6.11. In particular, notice the unreasonable initial values as indicated by the vertical scales which were increased by a factor of 5 to 10 over the calm air scales in Figure 6.5.



- Notes: 1. Flight Condition: 4,000 ft PA, 0.65 M, Turbulent Air
 2. cg location: 36.85 percent mean aerodynamic chord

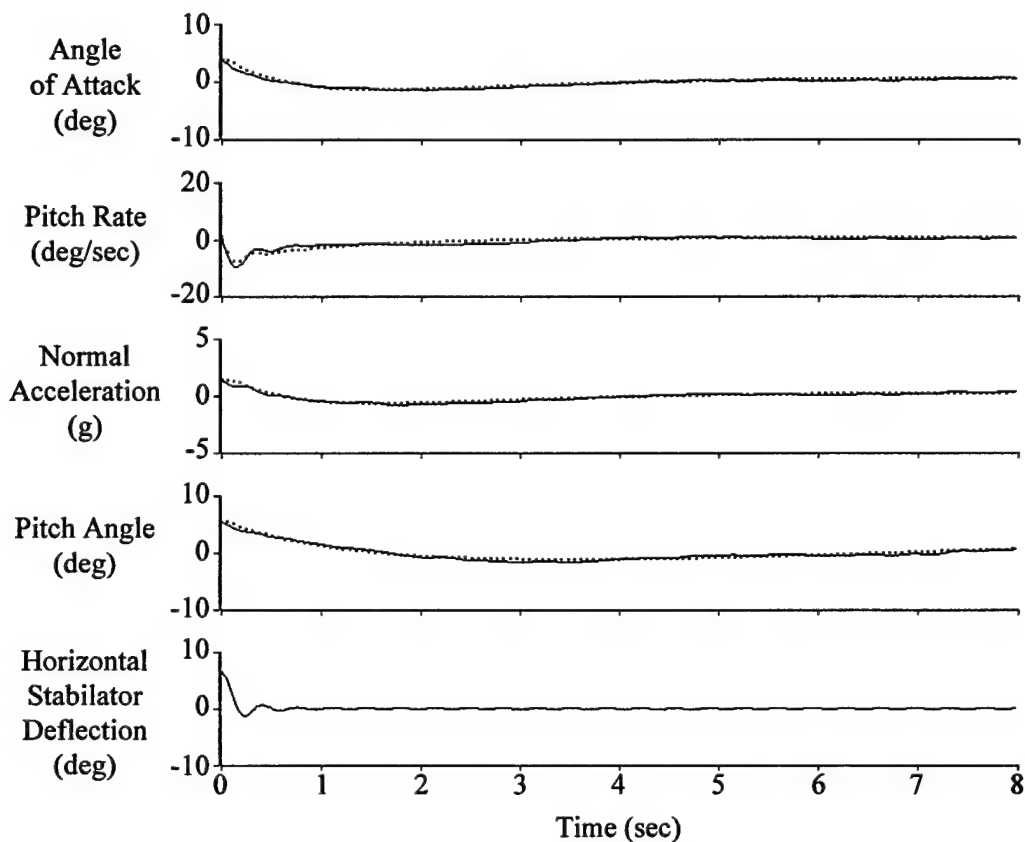
Figure 6.11. F-16B Turbulent Air Discrete Pulse Responses From the Have Derivatives Process

The responses in Figure 6.11 were imported into PCPID along with the simulated ideal impulse input. The resulting time history matches were practically useless as PCPID attempted to correlate the excessive initial values of the discrete pulse responses with the simulated ideal impulse input. Unreasonable stability derivative estimates were computed by PCPID in conjunction with the very poor time history matches. Finally, excessive Cramer-Rao bounds completed the unfiltered turbulent air results from the Have Derivatives process with PCPID.

The extremely poor results indicated that the Have Derivatives process was ineffective at minimizing the noise degradation. Perhaps a broadband input would have enabled the ensemble

averaging step of the process to minimize the variance as it had in simulation, giving much better turbulent air stability derivative estimation results.

The same low-pass Butterworth filters were applied to the turbulent air data, using MATLAB's® `FILTFILT.M` routine, as was done with the calm air data. Again, the simulated ideal impulse input was filtered independently and imported into PCPID along with the filtered discrete pulse responses. Time history matches of the filtered turbulent air signals are presented in Figure 6.12.



..... PCPID Computed Response — Measured Aircraft Response

- Notes: 1. PCPID - Personal Computer Parameter Identification
 2. Flight Condition: 4,000 ft PA, 0.65 M, Turbulent Air
 3. cg location: 36.85 percent mean aerodynamic chord

Figure 6.12. Matches of PCPID Computed Time Histories to Filtered F-16B Turbulent Air Discrete Pulse Responses From the Have Derivatives Process

The time history matches in Figure 6.12 were very good considering how degraded the signals were prior to filtering, as shown earlier in Figure 6.11. As in the calm air example, the low-pass Butterworth filters were able to eliminate most of the high frequency noise. Despite the greatly increased degradation present in Figure 6.11, as compared to the unfiltered calm air responses in Figure 6.5, the filtered turbulent air responses in Figure 6.12 were almost identical to the filtered calm air signals in Figure 6.7. Those qualitative comparisons illustrated well the effectiveness of the low-pass filters.

To quantify the low-pass filters' effectiveness, comparisons are made in Table 6.11 between filtered turbulent air results from the Have Derivatives process with PCPID, filtered calm air results from the Have Derivatives process with PCPID and the flight test basis data set results.

Table 6.11. F-16B Filtered Results From the Have Derivatives Process With PCPID and Associated Flight Test Basis Data Set Results

Stability Derivative	Basis Data Set Calm Air, PCPID Alone ^{1,2}		Filtered, Calm Air, Have Derivatives with PCPID ^{3,4}		Filtered, Turbulent Air, Have Derivatives with PCPID ^{4,5}	
	Mean Estimate	Mean Cramer- Rao Bound	Estimate	Cramer- Rao Bound	Estimate	Cramer- Rao Bound
$C_{N\alpha}$ (per deg)	0.0714	$\pm 4.4\text{E-}03$	0.0661	$\pm 4.2\text{E-}03$	0.0639	$\pm 6.2\text{E-}03$
$C_{N\delta e}$ (per deg)	0.0132	$\pm 4.8\text{E-}03$	0.0182	$\pm 5.1\text{E-}03$	0.0192	$\pm 8.1\text{E-}03$
$C_{m\alpha}$ (per deg)	0.0008	$\pm 5.7\text{E-}04$	0.0007	$\pm 1.1\text{E-}04$	0.0001	$\pm 1.4\text{E-}04$
$C_{m\dot{\alpha}}$ (per rad)	-3.1380	$\pm 1.1\text{E-}00$	-7.0223	$\pm 3.9\text{E-}01$	-3.9577	$\pm 6.2\text{E-}01$
$C_{m\delta e}$ (per deg)	-0.0114	$\pm 9.9\text{E-}04$	-0.0132	$\pm 5.3\text{E-}04$	-0.0104	$\pm 9.0\text{E-}04$

- Notes: 1. PCPID - Personal Computer Parameter Identification
2. Flight condition: 4,000 ft PA, 0.65 M
3. Cramer-Rao bounds have been multiplied by a factor of 10

¹cg range: 37.38 to 36.15 percent mean aerodynamic chord

²Based on 32 samples

³cg location: 37.03 percent mean aerodynamic chord

⁴Based on 1 sample

⁵cg location: 36.85 percent mean aerodynamic chord

The results presented in Table 6.11 were mixed. All of the filtered turbulent air Have Derivatives Cramer-Rao bounds were larger than the filtered calm air Have Derivatives results, as expected. However, an unexpected finding was that the filtered turbulent air results showed smaller Cramer-Rao bounds than the associated basis data set bounds for $C_{m\alpha}$, C_{mq} and $C_{m\delta e}$.

Another unexpected outcome was that the filtered turbulent air Have Derivatives estimates for C_{mq} and $C_{m\delta e}$ were closer to their basis data set mean estimates than filtered calm air Have Derivatives values for the same derivatives. Based the results, the low-pass filters appeared to be effective at minimizing the estimation errors due to process and measurement noises.

A final set of numerical results are presented in Table 6.12. Filtered turbulent air results using the Have Derivatives process with PCPID are quantitatively compared to associated basis data set results and to turbulent air doublet results using PCPID alone.

Table 6.12. F-16B Filtered Turbulent Air Results From the Have Derivatives Process With PCPID, Turbulent Air Doublet Results From PCPID Alone, and Associated Flight Test Basis Data Set Results

Stability Derivative	Basis Data Set Calm Air PCPID Alone ^{1,2}		Doublet Input, Turbulent Air, PCPID Alone ^{3,4}		Filtered, Turbulent Air Have Derivatives with PCPID ^{3,4}	
	Mean Estimate	Mean Cramer-Rao Bound	Estimate	Cramer-Rao Bound	Estimate	Cramer-Rao Bound
$C_{N\alpha}$ (per deg)	0.0714	$\pm 4.4\text{E-}03$	0.0704	$\pm 3.9\text{E-}03$	0.0639	$\pm 6.2\text{E-}03$
$C_{N\delta e}$ (per deg)	0.0132	$\pm 4.8\text{E-}03$	0.0179	$\pm 5.4\text{E-}03$	0.0192	$\pm 8.1\text{E-}03$
$C_{m\alpha}$ (per deg)	0.0008	$\pm 5.7\text{E-}04$	0.0008	$\pm 3.0\text{E-}04$	0.0001	$\pm 1.4\text{E-}04$
C_{mq} (per rad)	-3.1380	$\pm 1.1\text{E-}00$	-4.1945	$\pm 8.2\text{E-}01$	-3.9577	$\pm 6.2\text{E-}01$
$C_{m\delta e}$ (per deg)	-0.0114	$\pm 9.9\text{E-}04$	-0.0109	$\pm 8.3\text{E-}04$	-0.0104	$\pm 9.0\text{E-}04$

- Notes: 1. PCPID - Personal Computer Parameter Identification
2. Flight condition: 4,000 ft PA, 0.65 M
3. Cramer-Rao bounds have been multiplied by a factor of 10

¹cg range: 37.38 to 36.15 percent mean aerodynamic chord ²Based on 32 samples

³cg location: 37.03 percent mean aerodynamic chord ⁴Data from Flight 5, Pass 5

Again the results were mixed. While the filtered turbulent air Have Derivatives results in Table 6.11 appeared to indicate significant benefit from using the low-pass filters, the turbulent air doublet results in Table 6.12 were better in most cases. The turbulent air doublet estimates were closer to the associated flight test basis data set mean estimates for all derivatives except C_{mq} . In addition, the turbulent air doublet Cramer-Rao bounds were smaller than the filtered turbulent air Have Derivatives Cramer-Rao bounds except for $C_{m\alpha}$ and C_{mq} .

While the comparisons from Table 6.11 appeared to indicate significant benefit from using the low-pass Butterworth filters in conjunction with the Have Derivatives process, the comparisons based on results in Table 6.12 indicated that equal or better performance was achieved using the current method of processing doublet input and responses with PCPID alone.

Based on those observations, for signals used in parameter estimation, it is again recommended that low-pass filters not be used to remove noise unless absolutely necessary, and then only at a cutoff frequency high enough above the dynamics of interest so as not to corrupt those dynamics.

Overall, the best flight data results were achieved using the current method in calm air. Results using the Have Derivatives process with PCPID should improve with a uniformly distributed frequency content input, such as the broadband input used in simulation. If flight results approach the Have Derivative process performance realized in simulation, the process should give better stability derivative estimation results in both calm and turbulent conditions.

6.7 Summary

A flight test program was executed in accordance with the Project HAVE DERIVATIVES test plan with the following conclusions:

1. USAF Test Pilot School data reduction processes and test aircraft instrumentation were validated using results presented in AFFTC-TLR-92-12.
2. A basis data set of calm air stability derivative estimation results was established with high confidence using doublet input and response signals in PCPID alone.
3. As expected, turbulent air results using doublet input and response signals with PCPID alone were consistently worse than calm air doublet results using PCPID alone.
4. The Have Derivatives process was validated with an F-16B simulation model, achieving excellent results similar to earlier AT-37B simulations.
5. Calm and turbulent air results using the Have Derivatives process with PCPID were unacceptable due to high frequency noise on the discrete pulse responses.
6. The ensemble averaging step of the Have Derivatives process was not effective enough at removing the signal noise.
7. Low-pass Butterworth filters applied forward and reverse in time were very effective at minimizing high frequency signal noise while inducing zero phase error.
8. Low-pass filters applied to the response signals before Have Derivatives processing, and independently to a simulated ideal impulse input signal, resulted in smaller Cramer-Rao bounds when incorporated into the Have Derivatives process with PCPID.
9. Variable amplitude attenuation caused by the low-pass filters resulted in unacceptable estimate accuracy.

7. Conclusions and Recommendation

7.1 Overview

Conclusions from simulation and flight test results are repeated in this chapter to summarize the overall research findings. Based on the conclusions, a recommendation is made for continued evaluation of the Have Derivatives process.

7.2 Simulation Conclusions

The following conclusions are drawn based on the simulation stability derivative estimation results from PCPID alone and from the Have Derivatives process with PCPID:

1. A simulation baseline set of stability derivative estimation results was established from simulated data with no noise, using a doublet input and PCPID alone.
2. Increasing process and / or measurement noise levels resulted in less accurate stability derivative estimates.
3. The output error parameter estimation algorithm in PCPID performed best when given the impulse response.
4. The optimum amount of section overlap was 50% for the given conditions of 1024 point section length and 2048 point maneuver length.
5. For a maneuver length of 2048 points and 50% section overlap, best results were achieved with a section length of 1024 points.
6. A maneuver length of 2048 points provided the best balance of accuracy and economy.

7.3 Flight Test Conclusions

A flight test program was executed in accordance with the Project HAVE DERIVATIVES test plan presented in Chapter 5 with the following conclusions:

1. USAF Test Pilot School data reduction processes and test aircraft instrumentation were validated against results presented in AFFTC-TLR-92-12.
2. A flight test basis set of calm air stability derivative estimation results was established with high confidence using doublet input and response signals in PCPID alone.
3. As expected, turbulent air results using doublet input and response signals with PCPID alone were consistently worse than calm air doublet results using PCPID alone.
4. The Have Derivatives process was validated with an F-16B simulation model, achieving excellent results similar to earlier AT-37B simulations.
5. Calm and turbulent air results using the Have Derivatives process with PCPID were unacceptable due to high frequency noise on the discrete pulse responses.
6. The ensemble averaging step of the Have Derivatives process was not effective enough at removing the signal noise.
7. Low-pass Butterworth filters applied forward and reverse in time were very effective at minimizing high frequency signal noise while inducing zero phase error.
8. Low-pass filters applied to the response signals before Have Derivatives processing, and independently to a simulated ideal impulse input signal, resulted in smaller Cramer-Rao bounds when incorporated into the Have Derivatives process with PCPID.
9. Variable amplitude attenuation caused by the low-pass filters resulted in unacceptable estimate accuracy.

7.4 Recommendation for Further Research

The Have Derivatives process performed very well in simulation and should be evaluated further in flight test. However, a broadband input, applied through a programmed test input capability, should be used as the flight test input. A broadband input would provide the uniform frequency content required by the ensemble averaging step of the Have Derivatives process. With a broadband input, the Have Derivatives process should dramatically improve stability derivative estimation results from both calm and turbulent air data.

Appendix A : Have Derivatives Simulation Program

The complete code for the simulation program developed in Chapter 3 is presented in this appendix. Comments within the code explain each section. The program was written and run using MATLAB®, version 4.2c.1 and the Signal Processing Toolbox®, version 3.0b.

This particular version of the program was modified for the F-16B that was used for flight test. The earlier version contained AT-37B data supplied by the USAF Test Pilot School.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Have Derivatives Simulator (HD_SIM.M)
%
%   Written by:  Capt Lars Hoffman           Version 5.0           1 October 1997
%
%   This script M-file creates continuous, linear, time-invariant, state space A, B, C, and D
%   matrices in physical variable form based on the short period approximation aircraft
%   equations of motion and user supplied constant and parameter inputs.
%
%   Linear simulations using this model and user defined input are then accomplished both
%   with and without the Have Derivatives process (ensemble averaging).
%
%   The output from these simulations are saved in GetData format for input into PCPID.
%   Additionally, various plots are drawn to aid analysis of results.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all           % Close all figure windows
clear all          % Clear MATLAB® memory
clc                % Clear command window

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   The first section builds the aircraft model and inputs, including turbulence and
%   measurement noise. The noise levels can be adjusted by the user.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%%%%% Aircraft Type: F-16B

<u>% VARIABLE</u>	<u>DESCRIPTION</u>	<u>UNITS</u>
-------------------	--------------------	--------------

%%%%% Average Flight Conditions During Maneuver

alt = 4000.0;	% average altitude	[ft]
avg_qbar = 540.0;	% average dynamic pressure	[lbf/ft ²]
avg_mach = 0.65;	% average mach number	[nondimensional]
avg_v = 716.0;	% average velocity	[ft/sec]
g = 32.2;	% gravitational acceleration	[ft/sec ²]

%%%%% Average Aircraft Mass Distribution

%

% These values were given by MASSCALC for a total fuel weight of 4000 lb. This was typical
% for half way through a mission based on a normal F-16B fuel burn sequence.

mass = 695.93;	% aircraft mass	[slugs]
ix = 8205.0;	% aircraft roll inertia	[slug-ft ²]
iy = 61237.0;	% aircraft pitch inertia	[slug-ft ²]
iz = 61509.0;	% aircraft yaw inertia	[slug-ft ²]
ixy = 0.0;	% aircraft X-Y product of inertia	[slug-ft ²]
ixz = 113.0;	% aircraft Y-Z product of inertia	[slug-ft ²]
iyz = 0.0;	% aircraft Y-Z product of inertia	[slug-ft ²]

%%%%% Aircraft Reference Dimensions

area = 300.0;	% aircraft reference area	[ft ²]
span = 30.0;	% aircraft reference span	[ft]
chord = 11.32;	% aircraft reference chord	[ft]

%%%%% Aircraft Center of Gravity Location

%

% These values were calculated based on MASSCALC adjustments to reference cg location for
% the total fuel weight of 4000 lb.

xcg = 26.92;	% X-coordinate of cg	[ft]
ycg = 0.01;	% Y-coordinate of cg	[ft]
zcg = 7.62;	% Z-coordinate of cg	[ft]

%%%%% Predicted Values for Longitudinal Stability and Control Derivatives

%

% These were the mean values from 32 calm air runs using doublet inputs and PCPID alone,
% which is the current stability derivative estimation method. The only exception is pcma
% which was actually 0.0007. Eleven of the runs resulted in values of cma from -0.0001 to
% -0.0012, while the other 21 runs had cma from 0.0000 to 0.0045. The positive values were
% due to the fact that the bare airframe F-16B has almost neutral static longitudinal stability at
% this flight condition and configuration.

%

% Because a positive cma in this program's aircraft model would yield a divergent response
% from the linear simulation, the small negative value of -0.0001 was used for the simulations.

pcNorma = 0.70e-1; % normal force due to alpha [1/deg]
pcNormde = 0.10e-1; % normal force due to elevator [1/deg]

pcma = -0.10e-3; % pitching moment due to alpha [1/deg]
pcmqa = -3.0; % pitching moment due to pitch rate [1/rad]
pcmda = -0.10e-1; % pitching moment due to elevator [1/deg]

%%%%% Longitudinal Increments to Initial Conditions

alpha0 = 0.0; % increment to alpha initial condition [deg]
q0 = 0.0; % increment to pitch rate initial condition [deg/sec]

%%%%% Longitudinal Response Biases

alphaBias = 0.0; % measurement bias to angle of attack [deg]
qBias = 0.0; % measurement bias to pitch rate [deg/sec]
nzBias = 0.0; % measurement bias to normal acceleration [g]

%%%%% Longitudinal Instrumentation Corrections

%

% These values are referenced to flight station 0.0, aircraft centerline and ground. They were
% provided by the USAF TPS Instrumentation Branch and Mr. Chris Nagy of Quartic
% Engineering Inc.

ka = 1.0; % upwash factor for angle of attack sensor [nondimensional]
xa = -2.29; % X-coordinate of angle of attack sensor [ft]
ya = 0.0; % Y-coordinate of angle of attack sensor [ft]
za = 7.58; % Z-coordinate of angle of attack sensor [ft]
xnz = 6.87; % X-coordinate of normal accelerometer [ft]
ynz = -0.64; % Y-coordinate of normal accelerometer [ft]
znz = 8.13; % Z-coordinate of normal accelerometer [ft]

%%%%%%%% Conversion Factor

r2d = 180/pi; % conversion from radians to degrees [deg/rad]

%%%%%%%% Turbulence Model Constants

%

% These values were based on guidance found in MIL-STD-1797A and other references.

ow = 4; % root-mean-square turbulence amplitude [ft/sec]

Lw = 500; % turbulence scale length / 2 [ft]

%%%%%%%% Measurement Noise Filter State Space Model

nlg = .1; % over-all measurement noise level gain [nondimensional]

ag = 0.20; % alpha measurement uncertainty [deg]

qg = 0.10; % q measurement uncertainty [deg/sec]

nzg = 0.04; % nz measurement uncertainty [g]

deg = 0.15; % de measurement uncertainty [deg]

bfmo = 4; % Butterworth filter model order [nondimensional]

mnwn = 100; % measurement noise cutoff frequency [rad/sec]

[Amn,Bmn,Cmn,Dmn] = butter(bfmo,mnwn,'high','s');

%%%%%%%% Aircraft Longitudinal State Space Model Including Turbulence (Dryden form)

c1 = (r2d*avg_qbar*area) / (mass*avg_v);

c2 = (avg_qbar*area*chord) / (2*mass*avg_v^2);

c3 = (r2d*avg_qbar*area*chord) / iy;

c4 = (avg_qbar*area*chord^2) / (2*iy*avg_v);

c5 = (r2d*ow*sqrt(avg_v)) / (Lw*sqrt(8*pi*Lw));

c6 = (r2d*ow*sqrt(pi*avg_v^3)) / (8*span*Lw*sqrt(2*Lw));

c7 = (r2d*ow*sqrt(3)) / sqrt(2*pi*Lw*avg_v);

c8 = (r2d*ow*sqrt(3*avg_v*pi)) / (4*span*sqrt(2*Lw));

c9 = (r2d*avg_v^2*pi^2) / (16*span^2);

c10 = ((xcg - xnz)*chord) / iy;

ap11 = -c1*pcNorma;

ap13 = (c1*pcNorma*c5);

ap14 = (c1*pcNorma*c7);

ap21 = c3*pcma;

ap22 = c4*pcmq;

ap23 = (-c3*pcma*c5) - (c4*pcmq*c6);

ap24 = (-c3*pcma*c7) - (c4*pcmq*c8);

ap25 = -c4*pcmq*c9;

```

ap43 = -(avg_v^2) / (4*Lw^2);
ap44 = -avg_v / Lw;
ap53 = -c5;
ap54 = -c7;
ap55 = -(r2d*avg_v*pi) / (4*span);

```

```

Ap = [ ap11      1      0      ap14      0;
       ap21      ap22    ap23      ap24    ap25;
       0         0      0         1        0;
       0         0      ap43      ap44      0;
       0         0      ap53      ap54      ap55];

```

```

bp11 = -c1*pcNormde;
bp21 = c3*pcmde;

```

```

Bp = [ bp11      0      0      0      0;
       bp21      0      0      0      0;
       0         0      0      0      0;
       0         1      0      0      0;
       0         0      0      0      0];

```

```

cp11 = ka;
cp12 = (ka*(xa-xcg)) / avg_v;
cp31 = (((avg_qbar*area) / g) * (pcNorma / mass)) + (((avg_qbar*area) / g) * (c10*pcma));
cp32 = ((avg_qbar*area*chord)/(2*g*avg_v*r2d))*(c10*pcm);

```

```

Cp = [ cp11      cp12      0      0      0;
       0         1        0      0      0;
       cp31      cp32      0      0      0;
       0         0      -c5     -c7      0;
       0         0      -c6     -c8     -c9];

```

```

dp31 = (((avg_qbar*area) / g) * (pcNormde / mass)) + (((avg_qbar*area) / g) * (c10*pcmde));

```

```

Dp = [ 0         0      1      0      0;
       0         0      0      1      0;
       dp31      0      0      0      1;
       0         0      0      0      0;
       0         0      0      0      0];

```

%%%%%% Measured Values of Longitudinal State Parameters at Initial Point

```

alphaz0 = 0;           % measured angle of attack at initial point [deg]
qz0 = 0;               % measured pitch rate at initial point [deg/sec]

```

%%%%%%%% Initial Conditions

```
alpha_i = ((alphaz0 - alphaBias) / ka) + alpha0;
q_i = qz0 - qBias + q0;
x0 = [alpha_i; q_i; 0; 0; 0];
```

%%%%%%%% Time Vector

```
ml = 4096; % Maneuver length

t = (0 : 1/67 : (ml*(1/67))-(1/67)); % Time vector
```

%%%%%%%% Input Matrices

```
simulated_ideal_impulse = [zeros(100,1); 67.*ones(1,1); zeros(length(t)-101,1)];
doublet = [zeros(100,1); ones(50,1); -ones(50,1); zeros(length(t)-200,1)];
3211 = [zeros(100,1); ones(60,1); -ones(40,1); ones(20,1); -ones(20,1); zeros(length(t)-240,1)];
sine_sweep = [zeros(100,1); sin(t.*(0.5*t)).*ones(length(t),1)];
sine_dwell = [zeros(100,1); sin(t).*ones(length(t),1)];
```

```
randn('seed',0);
Gblwinpt = randn(length(t)-100,1);
zmGblwinpt = Gblwinpt - mean(Gblwinpt);
broad_band = [zeros(100,1); (1/std(zmGblwinpt)).*zmGblwinpt];
```

%%%%%%%% Zero-Mean, Gaussian Distributed, Band-Limited, Noise Vectors

```
randn('seed',10)
Gblwtpn = randn(length(t),1);
zmGblwtpn = Gblwtpn - mean(Gblwtpn);
zmGblwtpn = ow.*zmGblwtpn;
```

```
randn('seed',20);
Gblwamn = randn(length(t),1);
zmGblwamn = Gblwamn - mean(Gblwamn);
camn = lsim(Amn, Bmn, Cmn, Dmn, zmGblwamn, t);
csamn = nlq * (ag / std(camn)).*zmGblwamn;
```

```
randn('seed',30);
Gblwqmn = randn(length(t),1);
zmGblwqmn = Gblwqmn - mean(Gblwqmn);
cqmn = lsim(Amn, Bmn, Cmn, Dmn, zmGblwqmn, t);
csqmn = nlq * (qg / std(cqmn)).*zmGblwqmn;
```

```

randn('seed',40)
Gblwnzmn = randn(length(t),1);
zmGblwnzmn = Gblwnzmn - mean(Gblwnzmn);
cnzmn = lsim(Amn, Bmn, Cmn, Dmn, zmGblwnzmn, t);
csnzmn = nlsg * (nzg / std(cnzmn)).*zmGblwnzmn;

randn('seed',50)
Gblwdemn = randn(length(t),1);
zmGblwdemn = Gblwdemn - mean(Gblwdemn);
cdemn = lsim(Amn, Bmn, Cmn, Dmn, zmGblwdemn, t);
csdemn = nlsg * (deg / std(cdemn)).*zmGblwdemn;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                    %
%          This section accomplishes linear simulations using user defined inputs.          %
%                                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Unaveraged Responses for Comparison

% Unit Impulse Responses Without Noise

[ideal_impulse_responses] = [zeros(100,5); impulse(Ap, Bp, Cp, Dp, 1, t)];

% Responses Without Noise

u_sim_ideal = [doublet zeros(length(t),1) zeros(length(t),1) zeros(length(t),1) zeros(length(t),1)];

[y_sim_ideal] = lsim(Ap, Bp, Cp, Dp, u_sim_ideal, t, x0);

% Coherence Functions

C_sim_ideal(:,1) = cohere(u_sim_ideal(:,1),y_sim_ideal(:,1),fft1,67,[],0.5*fft1,'mean');
C_sim_ideal(:,2) = cohere(u_sim_ideal(:,1),y_sim_ideal(:,2),fft1,67,[],0.5*fft1,'mean');
C_sim_ideal(:,3) = cohere(u_sim_ideal(:,1),y_sim_ideal(:,3),fft1,67,[],0.5*fft1,'mean');

% Responses With Noise

u_turb_base = [doublet zmGblwtpn csamn csqmn csnzmn];

[y_turb_base] = lsim(Ap, Bp, Cp, Dp, u_turb_base, t, x0);

u_turb_base = u_turb_base(:,1) + csdemn(:,1);

```

% Coherence Functions

```
C_turb_base(:,1) = cohere(u_turb_base(:,1),y_turb_base(:,1),fft1,67,[],0.5*fft1,'mean');
C_turb_base(:,2) = cohere(u_turb_base(:,1),y_turb_base(:,2),fft1,67,[],0.5*fft1,'mean');
C_turb_base(:,3) = cohere(u_turb_base(:,1),y_turb_base(:,3),fft1,67,[],0.5*fft1,'mean');
```

%%%%%%%% Linear Simulation for Have Derivatives Process

```
u_turb = [broad_band zmGblwtpn csamn csqmn csnzmn];
```

```
[y_turb] = lsim(Ap, Bp, Cp, Dp, u_turb, t, x0);
```

```
u_turb = u_turb(:,1) + csdemn(:,1);
```

% Coherence Functions

```
C_turb(:,(j-1)*no+k) = cohere(u_turb,y_turb(:,(j-1)*no+k),fft1,67,[],ovlp,'mean');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   The Have Derivatives proces is accomplished in this section. This includes
%   computing averaged complex frequency response functions, mirroring those
%   functions about the Nyquist frequency and transforming them back into the time
%   domain as discrete pulse responses.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%%%%%%%% Ensemble Averaging Variables

```
ni = 1;           % number of command inputs (de)
no = 3;           % number of measured outputs (alpha, q, theta, nz)
fft1 = 1024;      % number of points per FFT
po = 50;          % percent overlap
ovlp = round(po*.01*fft1); % number of points overlapped
```

%%%%%%%% Frequency Response Estimation

```
for j = 1 : ni;
```

```
for k = 1 : no;
```

```
% Average Complex Frequency Response Function Estimates (Transfer Functions)
```

```
H(:,(j-1)*no+k) = tfe(u_turb,y_turb(:,(j-1)*no+k),fft1,67,[],ovlp,'mean');
```

```
% Complex Frequency Response Function Estimates Mirrored About the Nyquist Frequency
```

```
H_full(:,(j-1)*no+k) = mirror(H(:,(j-1)*no+k));
```

```
%%%%%%%%% Inverse Fast Fourier Transform Produced Discrete Pulse Responses
```

```
h_imp(:,(j-1)*no+k) = 67.*real(ifft(H_full(:,(j-1)*no+k)));
```

```
end
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                 %  
% Input, response and flight condition signals are defined in this section and saved %  
% in GetData format for input into PCPID. %  
%                                                                 %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
alpha = [zeros(100,1); (h_imp(1:fftl-100,1))];  
%alpha = y_sim_ideal(1:fftl,1);  
%alpha = y_turb_base(1:fftl,1);  
%alpha = ideal_impulse_response(1:fftl,1);
```

```
q = [zeros(100,1); (h_imp(1:fftl-100,2))];  
%q = y_sim_ideal(1:fftl,2);  
%q = y_turb_base(1:fftl,2);  
%q = ideal_impulse_response(1:fftl,2);
```

```
nz = [zeros(100,1); (h_imp(1:fftl-100,3))];  
%nz = y_sim_ideal(1:fftl,3);  
%nz = y_turb_base(1:fftl,3);  
%nz = ideal_impulse_response(1:fftl,3);
```

```
%de = u_sim_ideal(1:fftl,1);  
de_turb_base = u_turb_base(1:fftl,1);  
de = de_turb_base;
```

```
v = avg_v.*ones(fftl,1);  
alt = alt.*ones(fftl,1);  
mach = avg_mach.*ones(fftl,1);  
qbar = avg_qbar.*ones(fftl,1);
```

```
t = (0 : 1/67 : (fftl/67) - (1/67))';
```

%%%%%%%% Save Simulation Results in GetData Format

gdsave('C:\projects\Sim_Data\F16Dt3m0.gtd', t, alpha, q, nz, de, alt, v, mach, qbar);

%%
%
% Results are plotted by this final section to aid in analysis. %
%
%%

figure

subplot(4,1,1),plot(t(1:667,:),alpha(1:667:),'k-');
axis([0 10 -10 10]);
ylabel('alpha (deg)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);

subplot(4,1,2),plot(t(1:667,:),q(1:667:),'k-');
axis([0 10 -25 25]);
ylabel('q (deg/sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);

subplot(4,1,3),plot(t(1:667,:),nz(1:667:),'k-');
axis([0 10 -5 5]);
ylabel('nz (g)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);

subplot(4,1,4),plot(t(1:667,:),de(1:667:),'k-');
axis([0 10 -2 2]);
ylabel('de (deg)','FontName','Times','FontSize',11);
xlabel('Time (sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
input('');
figure

subplot(4,1,1),plot(t(1:667,:),y_sim_ideal(1:667,1),'k--',t(1:667,:),y_turb_base(1:667,1),'k-');
axis([0 10 -10 10]);
title(['Simulated Responses; No Noise (dashed), Noise(solid)'],'FontName','Times','FontSize',11);
ylabel('alpha (deg)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);

subplot(4,1,2),plot(t(1:667,:),y_sim_ideal(1:667,2),'k--',t(1:667,:),y_turb_base(1:667,2),'k-');
axis([0 10 -25 25]);
ylabel('q (deg/sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);

```
subplot(4,1,3),plot(t(1:667,:),y_sim_ideal(1:667,3),'k--',t(1:667,:),y_turb_base(1:667,3),'k-');
axis([0 10 -5 5]);
ylabel('nz (g)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
```

```
subplot(4,1,4),plot(t(1:667,:),de(1:667,:),'k--',t(1:667,:),de_turb_base(1:667,:),'k-');
axis([0 10 -2 2]);
ylabel('de (deg)','FontName','Times','FontSize',11);
xlabel('Time (sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
input("");
```

figure

```
subplot(4,1,1),plot(t(1:667,:),ideal_impulse_response(1:667,1),'k:');
axis([0 10 -10 10]);
title(['Ideal Impulse Responses'],'FontName','Times','FontSize',11);
ylabel('alpha (deg)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
```

```
subplot(4,1,2),plot(t(1:667,:),ideal_impulse_response(1:667,2),'k:');
axis([0 10 -25 25]);
ylabel('q (deg/sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
```

```
subplot(4,1,3),plot(t(1:667,:),ideal_impulse_response(1:667,3),'k:');
axis([0 10 -5 5]);
ylabel('nz (g)','FontName','Times','FontSize',11);
xlabel('Time (sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
input("");
```

```
figure,plot(t(1:333,:),alpha(1:333,:),'k-',t(1:333,:),y_sim_ideal(1:333,1),'k--',
    t(1:333,:),ideal_impulse_response(1:333,1),'k:');
axis([0 5 -10 10]);
title(['Measured" DPR(solid), Ideal DPR(--), Ideal Imp Resp(..)'],
    'FontName','Times','FontSize',11);
ylabel('alpha (deg)','FontName','Times','FontSize',11);
xlabel('Time (sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
input("");
```

```
figure,plot(t(1:333,:),q(1:333,:),'k-',t(1:333,:),y_sim_ideal(1:333,2),'k--',
    t(1:333,:),ideal_impulse_response(1:333,2),'k:');
axis([0 5 -25 25]);
title(['Measured" DPR(solid), Ideal DPR(--), Ideal Imp Resp(..)'],
    'FontName','Times','FontSize',11);
ylabel('q (deg/sec)','FontName','Times','FontSize',11);
xlabel('Time (sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
input("");
```

```
figure,plot(t(1:333,:),nz(1:333,:),'k-',t(1:333,:),y_sim_ideal(1:333,3),'k--',
    t(1:333,:),ideal_impulse_response(1:333,3),'k:');
axis([0 5 -5 5]);
title(['Measured" DPR(solid), Ideal DPR(--), Ideal Imp Resp(..)'],
    'FontName','Times','FontSize',11);
ylabel('nz (g)','FontName','Times','FontSize',11);
xlabel('Time (sec)','FontName','Times','FontSize',11);
set(gca,'FontName','Times','FontSize',11);
input("");
```

%%%%%%%% Ideal and Estimated Frequency Responses

```
w = (0 : (2*pi*67)/fft1 : (pi*67)); % Frequency vector to Nyquist Frequency [rad/sec]
[magi,phasei,w] = bode(Ap, Bp, Cp, Dp, 1, w);
```

figure

```
subplot(3,1,1),semilogx(w,20*log10(abs(H_full(1:length(w),1))), 'k-',w,20*log10(magi(:,1)), 'k:');
set(gca,'FontName','Times','FontSize',11);
title(['Estimated"(solid), Ideal(- -)']);
ylabel('alpha Magnitude (dB)','FontName','Times','FontSize',11);
```

```
subplot(3,1,2),semilogx(w,(180/pi).*angle(H_full(1:length(w),1)), 'k-',w,phasei(:,1), 'k:');
set(gca,'FontName','Times','FontSize',11);
ylabel('alpha Phase(deg)','FontName','Times','FontSize',11);
subplot(3,1,3),semilogx(w,C_turb_base(:,1), 'k-',w,C_sim_ideal(:,1), 'k:');
axis([.1 1000 0 1]);
set(gca,'FontName','Times','FontSize',11);
ylabel('alpha / de Coherence','FontName','Times','FontSize',11);
xlabel('Frequency (rad/sec)','FontName','Times','FontSize',11);
input("");
```

figure

```
subplot(3,1,1),semilogx(w,20*log10(abs(H_full(1:length(w),2))), 'k-',w,20*log10(magi(:,2)), 'k:');
set(gca,'FontName','Times','FontSize',11);
title(['"Estimated"(solid), Ideal(- -)']);
ylabel('q Magnitude (dB)','FontName','Times','FontSize',11);

subplot(3,1,2),semilogx(w,(180/pi).*angle(H_full(1:length(w),2))), 'k-',w,phasei(:,2), 'k:');
set(gca,'FontName','Times','FontSize',11);
ylabel('q Phase (deg)','FontName','Times','FontSize',11);

subplot(3,1,3),semilogx(w,C_turb_base(:,2), 'k-',w,C_sim_ideal(:,2), 'k:');
axis([.1 1000 0 1]);
set(gca,'FontName','Times','FontSize',11);
ylabel('q / de Coherence','FontName','Times','FontSize',11);
xlabel('Frequency (rad/sec)','FontName','Times','FontSize',11);
input("");
```

figure

```
subplot(3,1,1),semilogx(w,20*log10(abs(H_full(1:length(w),3))), 'k-',w,20*log10(magi(:,3)), 'k:');
set(gca,'FontName','Times','FontSize',11);
title(['"Measured"(solid), Ideal(- -)']);
ylabel('nz Magnitude (dB)','FontName','Times','FontSize',11);

subplot(3,1,2),semilogx(w,(180/pi).*angle(H_full(1:length(w),3))), 'k-',w,phasei(:,3), 'k:');
set(gca,'FontName','Times','FontSize',11);
ylabel('nz Phase (deg)','FontName','Times','FontSize',11);

subplot(3,1,3),semilogx(w,C_turb_base(:,3), 'k-',w,C_sim_ideal(:,3), 'k:');
axis([.1 1000 0 1]);
set(gca,'FontName','Times','FontSize',11);
ylabel('nz / de Coherence','FontName','Times','FontSize',11);
xlabel('Frequency (rad/sec)','FontName','Times','FontSize',11);
input("");
```

Appendix B : Simulation Results

Complete simulation results are contained in this appendix. The numerical results in the following spreadsheets were used to produce the bar charts in Chapter 4, Simulation Results and Analysis.

The deviations for each derivative were the difference between the estimated value and the actual value. The root-sum-of-squares magnitudes were calculated using Equation (4.1).

$$\text{RSS Magnitude} = \{ \sum [\text{derivative weighting} * (\text{estimate value} - \text{actual value})^2] \}^{\frac{1}{2}} \quad (4.1)$$

Table B.1. Process Noise Only Simulation Results

Derivatives	Actual	0 ft/sec	3 ft/sec	6 ft/sec	9 ft/sec
$C_{N\alpha}$	1.000E-01	1.001E-01	9.988E-02	9.234E-02	8.215E-02
Deviation	0.000E+00	1.220E-04	1.200E-04	7.660E-03	1.785E-02
$C_{N\delta_e}$	5.000E-03	5.621E-03	1.053E-02	2.449E-02	4.207E-02
Deviation	0.000E+00	6.210E-04	5.530E-03	1.949E-02	3.707E-02
$C_{m\alpha}$	-1.000E-02	-9.707E-03	-9.247E-03	-8.080E-03	-6.882E-03
Deviation	0.000E+00	2.930E-04	7.530E-04	1.920E-03	3.118E-03
C_{mq}	-2.000E+01	-1.916E+01	-2.150E+01	-2.380E+01	-2.629E+01
Deviation	0.000E+00	8.400E-01	1.500E+00	3.800E+00	6.290E+00
$C_{m\delta_e}$	-2.000E-02	-1.933E-02	-2.076E-02	-2.194E-02	-2.349E-02
Deviation	0.000E+00	6.700E-04	7.600E-04	1.940E-03	3.490E-03
RSS of Deviations	0.000E+00	1.878E+00	3.354E+00	8.497E+00	1.407E+01

Table B.2. Measurement Noise Only Simulation Results

Derivatives	Actual	0%	10%	20%	30%
$C_{N\alpha}$	1.000E-01	1.001E-01	1.016E-01	1.024E-01	1.024E-01
Deviation	0.000E+00	1.220E-04	1.600E-03	2.400E-03	2.400E-03
$C_{N\delta e}$	5.000E-03	5.621E-03	5.764E-03	5.603E-03	6.940E-03
Deviation	0.000E+00	6.210E-04	7.640E-04	6.030E-04	1.940E-03
$C_{m\alpha}$	-1.000E-02	-9.707E-03	-1.050E-02	-1.070E-02	-1.096E-02
Deviation	0.000E+00	2.930E-04	5.000E-04	7.000E-04	9.600E-04
C_{mq}	-2.000E+01	-1.916E+01	-1.837E+01	-1.776E+01	-1.664E+01
Deviation	0.000E+00	8.400E-01	1.630E+00	2.240E+00	3.360E+00
$C_{m\delta e}$	-2.000E-02	-1.933E-02	-1.970E-02	-1.953E-02	-1.915E-02
Deviation	0.000E+00	6.700E-04	3.000E-04	4.700E-04	8.500E-04
RSS of Deviations	0.000E+00	1.878E+00	3.645E+00	5.009E+00	7.513E+00

Table B.3. Combined Process and Measurement Noise Simulation Results

Derivatives	Actual	Ideal	Calm	Turbulent
Turbulence	N/A	0 ft/sec	3 ft/sec	9 ft/sec
Measurement Noise	N/A	0%	10%	30%
$C_{N\alpha}$	1.000E-01	1.001E-01	1.040E-01	9.823E-02
Deviation	0.000E+00	1.220E-04	4.000E-03	1.770E-03
$C_{N\delta e}$	5.000E-03	5.621E-03	5.332E-03	1.616E-02
Deviation	0.000E+00	6.210E-04	3.320E-04	1.116E-02
$C_{m\alpha}$	-1.000E-02	-9.707E-03	-9.614E-03	-8.493E-03
Deviation	0.000E+00	2.930E-04	3.860E-04	1.507E-03
C_{mq}	-2.000E+01	-1.916E+01	-2.135E+01	-2.414E+01
Deviation	0.000E+00	8.400E-01	1.350E+00	4.140E+00
$C_{m\delta e}$	-2.000E-02	-1.933E-02	-2.111E-02	-2.274E-02
Deviation	0.000E+00	6.700E-04	1.110E-03	2.740E-03
RSS of Deviations	0.000E+00	1.878E+00	3.019E+00	9.257E+00

Table B.4. No Noise Simulation Results Using Different Input Functions

Derivatives	Actual	Impulse	Doublet	3211	Broadband
$C_{N\alpha}$	1.000E-01	9.983E-02	1.001E-01	1.048E-01	9.668E-02
Deviation	0.000E+00	1.700E-04	1.220E-04	4.800E-03	3.320E-03
$C_{N\delta e}$	5.000E-03	4.998E-03	5.621E-03	2.984E-03	3.800E-03
Deviation	0.000E+00	2.000E-06	6.210E-04	2.016E-03	1.200E-03
$C_{m\alpha}$	-1.000E-02	-9.990E-03	-9.707E-03	-9.966E-03	-1.011E-02
Deviation	0.000E+00	1.000E-05	2.930E-04	3.400E-05	1.100E-04
C_{mq}	-2.000E+01	-2.005E+01	-1.916E+01	-1.836E+01	-2.116E+01
Deviation	0.000E+00	5.000E-02	8.400E-01	1.640E+00	1.160E+00
$C_{m\delta e}$	-2.000E-02	-2.073E-02	-1.933E-02	-1.926E-02	-2.041E-02
Deviation	0.000E+00	7.300E-04	6.700E-04	7.400E-04	4.100E-04
RSS of Deviations	0.000E+00	1.118E-01	1.878E+00	3.667E+00	2.594E+00

Table B.5. Percent Overlap Simulation Results

Derivatives	Actual	Baseline	0% Overlap	50% Overlap
$C_{N\alpha}$	1.000E-01	1.040E-01	7.860E-02	1.013E-01
Deviation	0.000E+00	4.000E-03	2.140E-02	1.300E-03
$C_{N\delta e}$	5.000E-03	5.332E-03	6.094E-03	4.918E-03
Deviation	0.000E+00	3.320E-04	1.094E-03	8.200E-05
$C_{m\alpha}$	-1.000E-02	-9.614E-03	-1.112E-02	-1.030E-02
Deviation	0.000E+00	3.860E-04	1.120E-03	3.000E-04
C_{mq}	-2.000E+01	-2.135E+01	-1.925E+01	-2.009E+01
Deviation	0.000E+00	1.350E+00	7.500E-01	9.000E-02
$C_{m\delta e}$	-2.000E-02	-2.111E-02	-1.898E-02	-1.891E-02
Deviation	0.000E+00	1.110E-03	1.020E-03	1.090E-03
RSS of Deviations	0.000E+00	3.019E+00	1.678E+00	2.013E-01

Table B.6. Ideal Conditions Section Length Simulation Results

Derivatives	Actual	Ideal Conditions Baseline	512 Section Length	1024 Section Length
$C_{N\alpha}$	1.000E-01	1.001E-01	1.156E-01	9.832E-02
Deviation	0.000E+00	1.220E-04	1.560E-02	1.680E-03
$C_{N\delta e}$	5.000E-03	5.621E-03	4.346E-03	5.088E-03
Deviation	0.000E+00	6.210E-04	6.540E-04	8.800E-05
$C_{m\alpha}$	-1.000E-02	-9.707E-03	-1.002E-02	-9.906E-03
Deviation	0.000E+00	2.930E-04	2.000E-05	9.400E-05
C_{mq}	-2.000E+01	-1.916E+01	-2.144E+01	-2.006E+01
Deviation	0.000E+00	8.400E-01	1.440E+00	6.000E-02
$C_{m\delta e}$	-2.000E-02	-1.933E-02	-1.999E-02	-1.992E-02
Deviation	0.000E+00	6.700E-04	1.000E-05	8.000E-05
RSS of Deviations	0.000E+00	1.878E+00	3.220E+00	1.343E-01

Table B.7. Calm Conditions Section Length Simulation Results

Derivatives	Actual	Calm Conditions Baseline	512 Section Length	1024 Section Length
$C_{N\alpha}$	1.000E-01	1.040E-01	1.258E-01	1.013E-01
Deviation	0.000E+00	4.000E-03	2.580E-02	1.300E-03
$C_{N\delta e}$	5.000E-03	5.332E-03	4.110E-03	4.918E-03
Deviation	0.000E+00	3.320E-04	8.900E-04	8.200E-05
$C_{m\alpha}$	-1.000E-02	-9.614E-03	-1.058E-02	-1.030E-02
Deviation	0.000E+00	3.860E-04	5.800E-04	3.000E-04
C_{mq}	-2.000E+01	-2.135E+01	-2.188E+01	-2.009E+01
Deviation	0.000E+00	1.350E+00	1.880E+00	9.000E-02
$C_{m\delta e}$	-2.000E-02	-2.111E-02	-1.917E-02	-1.891E-02
Deviation	0.000E+00	1.110E-03	8.300E-04	1.090E-03
RSS of Deviations	0.000E+00	3.019E+00	4.205E+00	2.013E-01

Table B.8. Turbulent Conditions Section Length Simulation Results

Derivatives	Actual	Turbulent Conditions Baseline	512 Section Length	1024 Section Length
$C_{N\alpha}$	1.000E-01	9.823E-02	1.397E-01	9.810E-02
Deviation	0.000E+00	1.770E-03	3.970E-02	1.900E-03
$C_{N\delta e}$	5.000E-03	1.616E-02	3.587E-03	4.840E-03
Deviation	0.000E+00	1.116E-02	1.413E-03	1.600E-04
$C_{m\alpha}$	-1.000E-02	-8.493E-03	-1.068E-02	-1.049E-02
Deviation	0.000E+00	1.507E-03	6.800E-04	4.900E-04
C_{mq}	-2.000E+01	-2.414E+01	-2.263E+01	-2.109E+01
Deviation	0.000E+00	4.140E+00	2.630E+00	1.090E+00
$C_{m\delta e}$	-2.000E-02	-2.274E-02	-1.860E-02	-1.830E-02
Deviation	0.000E+00	2.740E-03	1.400E-03	1.700E-03
RSS of Deviations	0.000E+00	9.257E+00	5.882E+00	2.437E+00

Table B.9. Maneuver Length Simulation Results

Derivatives	Actual	Turbulent Conditions Baseline	2048 Maneuver Length	4096 Maneuver Length	8192 Maneuver Length
$C_{N\alpha}$	1.000E-01	9.823E-02	1.397E-01	1.370E-01	1.004E-01
Deviation	0.000E+00	1.770E-03	3.970E-02	3.700E-02	4.000E-04
$C_{N\delta e}$	5.000E-03	1.616E-02	3.587E-03	3.413E-03	4.903E-03
Deviation	0.000E+00	1.116E-02	1.413E-03	1.587E-03	9.700E-05
$C_{m\alpha}$	-1.000E-02	-8.493E-03	-1.068E-02	-1.234E-02	-1.104E-02
Deviation	0.000E+00	1.507E-03	6.800E-04	2.340E-03	1.040E-03
C_{mq}	-2.000E+01	-2.414E+01	-2.263E+01	-1.900E+01	-1.938E+01
Deviation	0.000E+00	4.140E+00	2.630E+00	1.000E+00	6.200E-01
$C_{m\delta e}$	-2.000E-02	-2.274E-02	-1.860E-02	-1.887E-02	-1.928E-02
Deviation	0.000E+00	2.740E-03	1.400E-03	1.130E-03	7.200E-04
RSS of Deviations	0.000E+00	9.257E+00	5.882E+00	2.239E+00	1.386E+00

Appendix C : Data Reduction Programs

C.1 Have Derivatives Data Reduction Program

The complete code for the Have Derivatives data reduction program, HDPROCES.M, is presented in the first section of this appendix. The program was designed to be interactive, prompting the user for most of the steps. Despite this there was still four pages of instructions written in step-by-step detail to guide the uninitiated through the process of getting stability derivative estimation results from raw, DAS tape recorded, flight test data. Although there was a DAS malfunction on Flight 11, no data were lost or corrupted due to data reduction during the entire flight test program.

HDPROCES also offered several options that had to be set internally to the program before starting a data reduction run. The settings in the program as it is presented in this appendix were typical from the last few data reduction runs of the flight test program.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% HAVE DERIVATIVES DATA REDUCTION (hdproces.m)          15 Sept 97  %
%
% This MATLAB® script file was written by the Project HAVE DERIVATIVES team to %
% automate the data reduction process from the point of having flight data files in tab %
% delimited flat ASCII format to the GetData format that could be processed by PCPID. %
%
% HDPROCES is interactive. First the user is asked to select the desired flight data file %
% and ensure that its columns are correctly defined. The user is then asked to define the %
% acceptable data limits which are used to remove unacceptable data from the file. Next %
% the user interactively divides the file into the set of three doublets and the pitch %
% frequency sweep. The doublets portion of the file is saved in the GetData format and is %
% available for PCPID processing. %
%
% The sweep portion is further processed through the Have Derivatives process by %
```

```

% transforming the input and response signals into the frequency domain via %
% MATLAB's® FFT routine, ensemble averaging the signals in the frequency domain, and %
% estimating the average complex frequency response functions using the transfer function %
% estimate (TFE.M) routine in MATLAB's® Signal Processing Toolbox®. Next, the %
% average complex frequency response function estimates are modified using the %
% MIRROR.M routine to fold the real parts and fold and flip the imaginary parts about the %
% Nyquist frequency to properly format the functions before inverse transforming. The %
% full average complex frequency response function estimates are then transformed back %
% into the time domain with IFFT where they become the system's discrete pulse %
% responses. These discrete pulse responses and a simulated ideal impulse input are saved %
% in the GetData format, as before, for PCPID processing. %
% %
% HDPROCES also prompts the user to define the trim shot period from which the %
% program calculates the average normal acceleration variation at the approximate short %
% period undamped natural frequency of 1 rad/sec. This value is hand recorded and used %
% to quantify the turbulence level for that run based on Have Derivatives defined %
% turbulence categories. %
% %
% Finally, HDPROCES calls MASSCALC, a FORTRAN program written by Mr. Chris %
% Nagy of Quartic Engineering Inc.. MASSCALC computes the mass and inertia %
% properties of F-16B, S/N 80-0635, given the total fuel weight. The output, including %
% mass, cg location and moments and products of inertia, are hand recorded and used as %
% input to PCPID. %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

close all      % Close all figure windows
clear all     % Clear MATLAB® memory
clc           % Clear command window

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% The first section has the user load the correct flight data file which is one flight data %
% collection pass consisting of a trim shot, a set of three doublets, another trim shot, and a %
% pitch frequency sweep. The user is asked to ensure that the data columns and acceptable %
% data limits are defined as desired. The acceptable data limits are defined to remove any %
% data that are clearly wrong in magnitude or sign (wild points) due to data acquisition %
% system transients. %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%% Load input file

```

```

[datfile,path] = uigetfile('*.','Choose input file. ');
nletters      = max(size(datfile));
varname       = datfile(1,1:nletters-4);

```

```
eval(['load ',path,datfile,'-ascii']);
input_file = eval(['varname '(:,:)']);
```

%%%%%%%% Column definition confirmation

```
disp("")
disp('Default Column Order is:')
disp("")
disp('Line          1')    % line number
disp('Time (s)      2')    % time          [sec]
disp('M_IC          3')    % Mach number
disp('QBAR          4')    % dynamic pressure [lb/ft²]
disp('V_T           5')    % true velocity  [ft/sec]
disp('ALPHA         6')    % angle of attack [deg]
disp('HORIZONTAL_TAIL_POSITION_RIGHT 7') % stabilator deflection [deg]
disp('N_Z           8')    % normal acceleration [g]
disp('Q             9')    % pitch rate    [deg/sec]
disp('H_IC          10')   % pressure altitude [ft]
disp('W_F           11')   % total fuel weight [lb]
disp('THETA         12')   % pitch angle    [deg]
disp("")
disp('Enter 1 if you wish to re-define the column order.')
column_order_answer = input('Enter 2 if you wish to go with the default values: ');
disp("")
```

```
if column_order_answer~=2
    input_file(1:5,:)
```

```
disp("")
actual_line = input('Type Line Number column # and press RETURN: ');
actual_time = input('Type Time column # and press RETURN: ');
actual_Mic = input('Type Mach column # and press RETURN: ');
actual_Qbar = input('Type Qbar column # and press RETURN: ');
actual_Vt = input('Type Vt column # and press RETURN: ');
actual_alpha = input('Type Alpha column # and press RETURN: ');
actual_Hstab = input('Type Horizontal Stab column # and press RETURN: ');
actual_Nz = input('Type Nz column # and press RETURN: ');
actual_Q = input('Type Q column # and press RETURN: ');
actual_Hic = input('Type Alt column # and press RETURN: ');
actual_Wf = input('Type Wf column # and press RETURN: ');
actual_theta = input('Type Theta column # and press RETURN: ');
disp("")
```

```
else
```

```
    actual_line = 1;
```

```

    actual_time    = 2;
    actual_Mic     = 3;
    actual_Qbar    = 4;
    actual_Vt      = 5;
    actual_alpha   = 6;
    actual_Hstab   = 7;
    actual_Nz      = 8;
    actual_Q       = 9;
    actual_Hic     = 10;
    actual_Wf      = 11;
    actual_theta   = 12;

end

clc                                % Clear command window

%%%%%% Acceptable data limit definition

disp('')
disp('Default acceptable data limits are:')
disp('')
disp('Line Number                [0, 1000000]')
disp('Time                      [time (s) in first line, time (s) in last line]')
disp('Mach Number                [0.6, 0.7]')
disp('Dynamic Pressure            [450, 650]')
disp('True Velocity               [600, 800]')
disp('Angle of Attack              [-2, 8]')
disp('Stabilator Deflection        [-10, 10]')
disp('Normal Acceleration          [-3, 4]')
disp('Pitch Rate                   [-20, 20]')
disp('Pressure Altitude            [3000, 5000]')
disp('Total Fuel Weight            [1000, 8000]')
disp('Pitch Angle                  [-10, 10]')
disp('')
disp('Enter 1 if you wish to define your own acceptable data limits.')
acceptable_data_limit_answer = input('Enter 2 if you wish to use the default values: ');
disp('')

if acceptable_data_limit_answer ~= 2

    disp('')
    Line      = input('Enter Min and Max value for Line Number: [min, max] ');
    Time      = input('Enter Min and Max value for Time: [min, max] ');
    M_IC      = input('Enter Min and Max value for Mach Number: [min, max] ');
    QBAR      = input('Enter Min and Max value for Dynamic Pressure: [min, max] ');
    V_T       = input('Enter Min and Max value for True Velocity: [min, max] ');

```

```

ALPHA    = input('Enter Min and Max value for Angle of Attack: [min, max] ');
HORIZONTAL_TAIL_POSITION_RIGHT = input('Enter Min and Max value for
                                         Stabilator Deflection: [min, max] ');
N_Z      = input('Enter Min and Max value for Normal Acceleration: [min, max] ');
Q        = input('Enter Min and Max value for Pitch Rate: [min, max] ');
H_IC     = input('Enter Min and Max value for Pressure Altitude: [min, max] ');
W_F      = input('Enter Min and Max value for Total Fuel Weight: [min, max] ');
THETA    = input('Enter Min and Max value for Pitch Angle: [min, max] ');
disp("")

else

    Line      = [0, 1000000];
    Time      = [input_file(1,actual_time), input_file(length(input_file),actual_time)];
    M_IC      = [0.6, 0.7];
    QBAR      = [450, 650];
    V_T       = [600, 800];
    ALPHA     = [-2, 8];
    HORIZONTAL_TAIL_POSITION_RIGHT = [-10, 10];
    N_Z       = [-3, 4];
    Q         = [-20, 20];
    H_IC      = [3000, 5000];
    W_F       = [1000, 8000];
    THETA     = [-10, 10];
end

clc                                % Clear command window

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Rows containing data beyond the acceptable limits defined in the last section are
% removed in this section.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Determine size of input file

[rows_input_file,columns_input_file] = size(input_file);

%%%%%%%%%% Remove rows containing non-numbers from input file

input_file(any(isnan(input_file')),:) = [];

```

%%%%%%%%%% Filter Line column

```
Good_Line_rows = find(input_file(:,actual_line)>=Line(1) &  
input_file(:,actual_line)<=Line(2));
```

```
input_file = input_file(Good_Line_rows,:);
```

%%%%%%%%%% Filter Time column

```
Good_Time_rows = find(input_file(:,actual_time)>=Time(1) &  
input_file(:,actual_time)<=Time(2));
```

```
input_file = input_file(Good_Time_rows,:);
```

%%%%%%%%%% Filter M_IC column

```
Good_M_IC_rows = find(input_file(:,actual_Mic)>=M_IC(1) &  
input_file(:,actual_Mic)<=M_IC(2));
```

```
input_file = input_file(Good_M_IC_rows,:);
```

%%%%%%%%%% Filter QBAR column

```
Good_QBAR_rows = find(input_file(:,actual_Qbar)>=QBAR(1) &  
input_file(:,actual_Qbar)<=QBAR(2));
```

```
input_file = input_file(Good_QBAR_rows,:);
```

%%%%%%%%%% Filter ALPHA column

```
Good_ALPHA_rows = find(input_file(:,actual_alpha)>=ALPHA(1) &  
input_file(:,actual_alpha)<=ALPHA(2));
```

```
input_file = input_file(Good_ALPHA_rows,:);
```

%%%%%%%%%% Filter HORIZONTAL_TAIL_POSITION_RIGHT column

```
Good_HORIZONTAL_TAIL_POSITION_RIGHT_rows =  
find(input_file(:,actual_Hstab)>=HORIZONTAL_TAIL_POSITION_RIGHT(1) &  
input_file(:,actual_Hstab)<=HORIZONTAL_TAIL_POSITION_RIGHT(2));
```

```
input_file = input_file(Good_HORIZONTAL_TAIL_POSITION_RIGHT_rows,:);
```

%%%%%%%%%% Filter N_Z column

```
Good_N_Z_rows = find(input_file(:,actual_Nz)>=N_Z(1) &  
                    input_file(:,actual_Nz)<=N_Z(2));
```

```
input_file = input_file(Good_N_Z_rows,:);
```

%%%%%%%%%% Filter Q column

```
Good_Q_rows = find(input_file(:,actual_Q)>=Q(1) & input_file(:,actual_Q)<=Q(2));
```

```
input_file = input_file(Good_Q_rows,:);
```

%%%%%%%%%% Filter H_IC column

```
Good_H_IC_rows = find(input_file(:,actual_Hic)>=H_IC(1) &  
                    input_file(:,actual_Hic)<=H_IC(2));
```

```
input_file = input_file(Good_H_IC_rows,:);
```

%%%%%%%%%% Filter V_T column

```
Good_V_T_rows = find(input_file(:,actual_Vt)>=V_T(1) &  
                    input_file(:,actual_Vt)<=V_T(2));
```

```
input_file = input_file(Good_V_T_rows,:);
```

%%%%%%%%%% Filter W_F column

```
Good_W_F_rows = find(input_file(:,actual_Wf)>=W_F(1) &  
                    input_file(:,actual_Wf)<=W_F(2));
```

```
input_file = input_file(Good_W_F_rows,:);
```

%%%%%%%%%% Filter THETA column

```
Good_THETA_rows = find(input_file(:,actual_theta)>=THETA(1) &  
                    input_file(:,actual_theta)<=THETA(2));
```

```
input_file = input_file(Good_THETA_rows,:);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Each of the input file signals is plotted versus time in this section for the user to ensure %
% that unacceptable data were removed. The time vector is first redefined to give a proper %
% time scale on the x-axis of each plot. The time step size is unaffected by the redefinition.%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Time vector re-definition

```

```

    input_file(:,actual_time)      = (0:1/67:(1/67)*length(input_file)-(1/67));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Plot Variables vs Time

```

```

figure

```

```

    Subplot(5,1,1);
    plot(input_file(:,actual_line),input_file(:,actual_time),'k-');
    set(gca,'FontName','Times','FontSize',20);
    title('PRESS ENTER TO CONTINUE','FontName','Times','FontSize',20);
    ylabel('Time (sec)','FontName','Times','FontSize',20);

```

```

    Subplot(5,1,2);
    plot(input_file(:,actual_time),input_file(:,actual_Mic),'k-');
    set(gca,'FontName','Times','FontSize',20);
    ylabel('Mic','FontName','Times','FontSize',20);

```

```

    Subplot(5,1,3);
    plot(input_file(:,actual_time),input_file(:,actual_Vt),'k-');
    set(gca,'FontName','Times','FontSize',20);
    ylabel('Vt (ft/sec)','FontName','Times','FontSize',20);

```

```

    Subplot(5,1,4);
    plot(input_file(:,actual_time),input_file(:,actual_Qbar),'k-');
    set(gca,'FontName','Times','FontSize',20);
    ylabel('Qbar (lb/ft^2)','FontName','Times','FontSize',20);

```

```

    Subplot(5,1,5);
    plot(input_file(:,actual_time),input_file(:,actual_Hic),'k-');
    set(gca,'FontName','Times','FontSize',20);
    xlabel('Line Number / Time (sec)','FontName','Times','FontSize',20);
    ylabel('Hic (ft)','FontName','Times','FontSize',20);
    input("");

```

figure

```
Subplot(5,1,1);
plot(input_file(:,actual_time),input_file(:,actual_alpha),'k-');
set(gca,'FontName','Times','FontSize',20);
title('PRESS ENTER TO CONTINUE','FontName','Times','FontSize',20);
ylabel('Alpha (deg)','FontName','Times','FontSize',20);
```

```
Subplot(5,1,2);
plot(input_file(:,actual_time),input_file(:,actual_Q),'k-');
set(gca,'FontName','Times','FontSize',20);
ylabel('Q (deg/sec)','FontName','Times','FontSize',20);
```

```
Subplot(5,1,3);
plot(input_file(:,actual_time),input_file(:,actual_Nz),'k-');
set(gca,'FontName','Times','FontSize',20);
ylabel('Nz (g)','FontName','Times','FontSize',20);
```

```
Subplot(5,1,4);
plot(input_file(:,actual_time),input_file(:,actual_theta),'k-');
set(gca,'FontName','Times','FontSize',20);
ylabel('Theta (deg)','FontName','Times','FontSize',20);
```

```
Subplot(5,1,5);
plot(input_file(:,actual_time),input_file(:,actual_Hstab),'k-');
set(gca,'FontName','Times','FontSize',20);
xlabel('Time (sec)','FontName','Times','FontSize',20);
ylabel('HStab (deg)','FontName','Times','FontSize',20);
input("");
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The user is now prompted, on a plot of Angle of Attack versus Time, to divide the data %
% file into two files, one containing the first trim shot and the set of three doublets, and the %
% other containing the second trim shot and the pitch frequency sweep. %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
[rows_new_input_file, columns_new_input_file] = size(input_file);
```

figure

```
plot(input_file(:,actual_time),input_file(:,actual_alpha),'k-');
set(gca,'FontName','Times','FontSize',20);
```

```

title('CLICK CROSSHAIRS AT END OF THIRD DOUBLET', 'FontName', 'Times',
      'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
ylabel('Alpha (deg)', 'FontName', 'Times', 'FontSize', 20);

split_time = ginput(1);
split_line = find(input_file(:,actual_time)>=split_time(1,1)-.01 &
                  input_file(:,actual_time)<=split_time(1,1)+.01);
doublet_input_file = input_file(1:split_line(1),:);

sweep_input_file = input_file(split_line(1):rows_new_input_file,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% On a plot of Normal Acceleration versus Time, the user is asked to define a trim
% shot period. HDPROCES calculates and plots the power spectral density of that time
% slice. The program also computes and displays the Normal Acceleration average
% deviation from 1 g at a frequency of 1 rad /sec. This value is hand recorded by the user
% for categorizing the turbulence level of this particular pass based on the definitions
% given in the body of this report.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Trim shot definition

figure

plot(input_file(:,actual_time), input_file(:,actual_Nz),'k-');
set(gca,'FontName','Times','FontSize',20);
title('CLICK CROSSHAIRS AT START AND THEN END OF TRIM SHOT', 'FontName',
      'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
ylabel('Nz (g)', 'FontName', 'Times', 'FontSize', 20);

[turbulence_X_time, turbulence_Y_time] = ginput(2);

turbulence_start_line = find(input_file(:,actual_time)>=turbulence_X_time(1)-.01 &
                             input_file(:,actual_time)<=turbulence_X_time(1)+.01);

turbulence_end_line = find(input_file(:,actual_time)>=turbulence_X_time(2)-.01 &
                            input_file(:,actual_time)<=turbulence_X_time(2)+.01);

```

%%%%%%%% Nz PSD calculation and plot

```
[Pxx,Freq] = psd(input_file(turbulence_start_line:turbulence_end_line,actual_Nz),  
[],67,[],[],'mean');
```

figure

```
plot(Freq,10*log10(Pxx),'k-');  
set(gca,'FontName','Times','FontSize',20);  
title('Nz PSD (PRESS ENTER TO CONTINUE)', 'FontName', 'Times', 'FontSize', 20);  
ylabel('Power Spectrum Magnitude (dB)', 'FontName', 'Times', 'FontSize', 20);  
xlabel('Frequency (Hz)', 'FontName', 'Times', 'FontSize', 20);  
input("");
```

%%%%%%%% Delta normal acceleration calculation

```
Pxx_at_1rad/sec = interp1(Freq,Pxx,1);
```

```
g_at_1rad/sec = sqrt(Pxx_at_1Hz*1);
```

```
Turbulence = g_at_1rad/sec
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                 %  
% Doublet signals are redefined in this section and saved in GetData format for PCPID. %  
%                                                                 %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Time = doublet_input_file(:,actual_time)-doublet_input_file(1,actual_time);
```

```
Mach = doublet_input_file(:,actual_Mic);
```

```
Qbar = doublet_input_file(:,actual_Qbar);
```

```
Alpha = doublet_input_file(:,actual_alpha);
```

```
Rstab = doublet_input_file(:,actual_Hstab);
```

```
Nz = doublet_input_file(:,actual_Nz);
```

```
Q = doublet_input_file(:,actual_Q);
```

```
Alt = doublet_input_file(:,actual_Hic);
```

```
Vt = doublet_input_file(:,actual_Vt);
```

```
Theta = doublet_input_file(:,actual_theta);
```

```
%%%%%%%% Save doublet signals in *.gtd format
```

```
gdsave('C:\projects\Calm_D\F3S04.gtd', Time, Alpha, Q, Nz, Theta, RStab, Mach, Qbar, Alt, Vt)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                 %  
% In this section, the pitch frequency sweep signals are redefined prior to Have Derivatives %  
% processing. Have Derivatives process variables and time and frequency vectors are also %  
% defined in this section.                                                                 %  
%                                                                 %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%% Redefined pitch frequency sweep vectors
```

```
Time = sweep_input_file(:,actual_time)-sweep_input_file(1,actual_time);
```

```
Mach = sweep_input_file(:,actual_Mic);
```

```
Qbar = sweep_input_file(:,actual_Qbar);
```

```
Alpha = sweep_input_file(:,actual_alpha);
```

```
Rstab = sweep_input_file(:,actual_Hstab);
```

```
Nz = sweep_input_file(:,actual_Nz);
```

```
Q = sweep_input_file(:,actual_Q);
```

```
Alt = sweep_input_file(:,actual_Hic);
```

```
Vt = sweep_input_file(:,actual_Vt);
```

```
Theta = sweep_input_file(:,actual_theta);
```

```
%%%%%%%% Defined Have Derivatives process variables
```

```
fftl = 1024;           % Section Length  
po = 50;               % Percent Overlap  
ovlp = round(po*.01*fftl); % Overlap Length
```

```
%%%%%%%% Time Vector [sec]
```

```
t = (0:1/67:(1/67)*fftl-(1/67));
```

```
%%%%%%%% Frequency Vectors [rad/sec]
```

```
frs = (0 : (2*pi*67)/fftl : (2*pi*67) - ((2*pi*67)/fftl)); % Full
```

```
nfrs = (0 : (2*pi*67)/fftl : (pi*67) - ((2*pi*67)/fftl)); % Only to Nyquist frequency
```

```
%%%%%%%%  
%  
% A low-pass Butterworth is created and applied in this section to minimize the high  
% frequency noise on the response signals prior to the frequency response estimation  
% step of the Have Derivatives process. First the unfiltered signals are plotted versus time.  
% Then the filter is applied to each response signal both forward and reverse in time to  
% induce zero phase lag. The filtered signals are then presented on the same plot as the  
% unfiltered signals to determine the effect of the filtering step.  
%  
%%%%%%%%
```

```
%%%%%%%% Plot of unfiltered pitch frequency sweep response signals
```

figure

```
subplot(5,1,1),plot(Time,Alpha,'k-');  
title('Unfiltered Pitch Frequency Sweep Responses vs Time', 'FontName', 'Times',  
      'FontSize', 20);  
ylabel('Alpha (deg)', 'FontName', 'Times', 'FontSize', 20);  
set(gca,'FontName','Times','FontSize',20,'xticklabels','');
```

```
subplot(5,1,2),plot(Time,Q,'k-');  
ylabel('Q (deg/sec)', 'FontName', 'Times', 'FontSize', 20);  
set(gca,'FontName','Times','FontSize',20,'xticklabels','');
```

```
subplot(5,1,3),plot(Time,Nz,'k-');  
ylabel('Nz (g)', 'FontName', 'Times', 'FontSize', 20);  
set(gca,'FontName','Times','FontSize',20,'xticklabels','');
```

```
subplot(5,1,4),plot(Time,Theta,'k-');  
ylabel('Theta (deg)', 'FontName', 'Times', 'FontSize', 20);  
set(gca,'FontName','Times','FontSize',20,'xticklabels','');
```

```
subplot(5,1,5),plot(Time,RStab,'k-');
```

```

ylabel('Stab Deflection (deg)', 'FontName', 'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca, 'FontName', 'Times', 'FontSize', 20);
input("");

```

%%%%%%%% Low-pass Butterworth filter creation and application

```

[b,a] = butter(9, 20/(pi*67)); % 9-pole filter with 20 rad/sec cutoff frequency

Alphaf = filtfilt(b,a,Alpha);

Qf = filtfilt(b,a,Q);

Nzf = filtfilt(b,a,Nz);

Thetaf = filtfilt(b,a,Theta);

```

%%%%%%%% Plots of unfiltered and filtered pitch frequency sweep responses

figure

```

plot(Time,Alpha,'r-',Time,Alphaf,'g:');
title('Unfiltered (solid red) and Filtered (dotted green) Responses vs Time', 'FontName',
      'Times', 'FontSize', 20);
ylabel('Alpha (deg)', 'FontName', 'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca, 'FontName', 'Times', 'FontSize', 20);
input("");

```

figure

```

plot(Time,Q,'r-',Time,Qf,'g:');
title('Unfiltered (solid red) and Filtered (dotted green) Responses vs Time', 'FontName',
      'Times', 'FontSize', 20);
ylabel('Q (deg/sec)', 'FontName', 'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca, 'FontName', 'Times', 'FontSize', 20);
input("");

```

figure

```

plot(Time,Nz,'r-',Time,Nzf,'g:');
title('Unfiltered (solid red) and Filtered (dotted green) Responses vs Time', 'FontName',
      'Times', 'FontSize', 20);
ylabel('Nz (g)', 'FontName', 'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca, 'FontName', 'Times', 'FontSize', 20);
input("");

```

figure

```
plot(Time,Theta,'r-',Time,Thetaf,'g:');
title('Unfiltered (solid red) and Filtered (dotted green) Responses vs Time', 'FontName',
      'Times', 'FontSize', 20);
ylabel('Theta (deg)', 'FontName', 'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);
input("");
```

%%%%%%%% Re-definition of filtered pitch frequency sweep response signals

Alpha = Alphaf;

Q = Qf;

Nz = Nzf;

Theta = Thetaf;

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                 %
% This section accomplishes the first two steps of the Have Derivatives process. Filtered %
% responses, the unfiltered input, and Have Derivatives variables defined above are given %
% as inputs to TFE.M, the MATLAB® Signal Processing Toolbox® routine that estimates %
% average complex frequency response functions using the algorithm described in 2.5.1 %
% of Chapter 2. The average complex frequency response function estimates returned by %
% TFE.M are modified by the MIRROR.M routine as described in 2.5.2 of Chapter 2. This %
% section also estimates the coherence function for each of the response and input %
% combinations. Finally, each of the full average complex frequency response function %
% estimates and coherence function estimates are plotted versus frequency, in rad/sec, out %
% to the Nyquist frequency, for user evaluation of the data quality. %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%%%%%%%% Full average complex frequency response and coherence function estimation

H_Alpha_Rstab = tfe(RStab,Alpha,fftl,67,[],ovlp,'mean');

H_Alpha_Rstab = mirror(H_Alpha_Rstab);

C_Alpha_Rstab = cohere(RStab,Alpha,fftl,67,[],ovlp, 'mean');

H_Q_Rstab = tfe(RStab,Q,fftl,67,[],ovlp, 'mean');

```

H_Q_Rstab      = mirror(H_Q_RStab);

C_Q_Rstab      = cohere(RStab,Q,fft1,67,[],ovlp, 'mean');

H_Nz_Rstab     = tfe(RStab,Nz,fft1,67,[],ovlp, 'mean');

H_Nz_Rstab     = mirror(H_Nz_RStab);

C_Nz_Rstab     = cohere(RStab,Nz,fft1,67,[],ovlp, 'mean');

H_Theta_Rstab  = tfe(RStab,Theta,fft1,67,[],ovlp, 'mean');

H_Theta_Rstab  = mirror(H_Theta_RStab);

C_Theta_Rstab  = cohere(RStab,Theta,fft1,67,[],ovlp, 'mean');

```

%% Plots of the average complex frequency response and coherence function estimates

figure

```

subplot(3,1,1),semilogx(nfrs,20*log10(abs(H_Alpha_RStab(1:length(nfrs),1))), 'k-')
title('Alpha / Horizontal Stabilator Average Frequency Response Function Estimate',
      'FontName', 'Times', 'FontSize', 20)
ylabel('Magnitude (dB)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20,'xticklabels',{''});

subplot(3,1,2),semilogx(nfrs,(180/pi).*angle(H_Alpha_RStab(1:length(nfrs),1)), 'k-');
ylabel('Phase (deg)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20,'xticklabels',{''});

subplot(3,1,3),semilogx(nfrs,C_Alpha_RStab, 'k-');
title('Alpha / Horizontal Stabilator Coherence Function Estimate', 'FontName', 'Times',
      'FontSize', 20)
ylabel('Coherence Function Estimate', 'FontName', 'Times', 'FontSize', 20);
xlabel('Frequency (rad / sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);
input('');

```

figure

```

subplot(3,1,1),semilogx(nfrs,20*log10(abs(H_Q_RStab(1:length(nfrs),1))), 'k-');
title('Q / Horizontal Stabilator Average Frequency Response Function Estimate',
      'FontName', 'Times', 'FontSize', 20);

```

```

ylabel('Magnitude (dB)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);

subplot(3,1,2),semilogx(nfrs,(180/pi).*angle(H_Q_RStab(1:length(nfrs),1)), 'k-');
ylabel('Phase (deg)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);
subplot(3,1,3),semilogx(nfrs,C_Q_RStab,'k-');
title('Q / Horizontal Stabilator Coherence Function Estimate', 'FontName', 'Times',
      'FontSize', 20);
ylabel('Coherence Function Estimate', 'FontName', 'Times', 'FontSize', 20);
xlabel('Frequency (rad / sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);
input("");

```

figure

```

subplot(3,1,1),semilogx(nfrs,20*log10(abs(H_Nz_RStab(1:length(nfrs),1))), 'k-');
title('Nz / Horizontal Stabilator Average Frequency Response Function Estimate',
      'FontName', 'Times', 'FontSize', 20);
ylabel('Magnitude (dB)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);

subplot(3,1,2),semilogx(nfrs,(180/pi).*angle(H_Nz_RStab(1:length(nfrs),1)), 'k-');
ylabel('Phase (deg)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);

subplot(3,1,3),semilogx(nfrs,C_Nz_RStab,'k-');
title('Nz / Horizontal Stabilator Coherence Function Estimate', 'FontName', 'Times',
      'FontSize', 20);
ylabel('Coherence Function Estimate', 'FontName', 'Times', 'FontSize', 20);
xlabel('Frequency (rad / sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);
input("");

```

figure

```

subplot(3,1,1),semilogx(nfrs,20*log10(abs(H_Theta_RStab(1:length(nfrs),1))), 'k-');
title('Theta / Horizontal Stabilator Average Frequency Response Function Estimate',
      'FontName', 'Times', 'FontSize', 20);
ylabel('Magnitude (dB)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);

subplot(3,1,2),semilogx(nfrs,(180/pi).*angle(H_Theta_RStab(1:length(nfrs),1)), 'k-');
ylabel('Phase (deg)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);

subplot(3,1,3),semilogx(nfrs,C_Theta_RStab,'k-');

```

```

title('Theta / Horizontal Stabilator Coherence Function Estimate', 'FontName', 'Times',
      'FontSize', 20);
ylabel('Coherence Function Estimate', 'FontName', 'Times', 'FontSize', 20);
xlabel('Frequency (rad / sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca, 'FontName', 'Times', 'FontSize', 20);
input("");

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The third and final step of the Have Derivatives process is accomplished in this section. %
% The full average complex frequency response functions are transformed back into the %
% time domain using MATLAB's® built-in inverse fast Fourier transform function. The %
% signals become discrete pulse responses in the time domain and are multiplied by the %
% sampling rate (67 samples per second in this case) so that they are a correct %
% approximation of the continuous inverse fast Fourier transform results. Because only the %
% initial part of the responses are important for PCPID processing, only the first 500 points %
% of each signal are saved in GetData format. That equates to approximately 7.5 seconds %
% at a 67 samples per second sampling rate. That was more than enough time to provide to %
% PCPID because the transient response died out in about 1.5 seconds in each case, leaving %
% the steady state response. Also created and saved with the response signals in this %
% section is the simulated ideal impulse response function. The simulated ideal impulse is %
% one time step wide and one over one time step in amplitude, which equated to an %
% amplitude of approximately 67 degrees. The same low-pass Butterworth filter applied %
% earlier to the response signals, prior to Have Derivatives processing, is applied here to %
% the input signal. Finally, the response and filtered simulated ideal impulse input signals %
% are saved in GetData format for PCPID processing. The signals, as saved, are plotted %
% versus time so the user may ensure that proper Have Derivatives processing was %
% accomplished. %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%% Inverse fast Fourier transformation and scaling

```

```

h_Alpha_RStab_impulse = 67.*(real(ifft(H_Alpha_RStab,fft1)));
h_Q_RStab_impulse     = 67.*(real(ifft(H_Q_RStab,fft1)));
h_Nz_RStab_impulse    = 67.*(real(ifft(H_Nz_RStab,fft1)));
h_Theta_RStab_impulse = 67.*(real(ifft(H_Theta_RStab,fft1)));

```

```

%%%%%%%%% Definition of clipped discrete pulse responses

```

```

h_Alpha_RStab_impulse_Clipped = [h_Alpha_RStab_impulse(1:500,:)];

```

```

h_Q_RStab_impulse_Clipped    = [h_Q_RStab_impulse(1:500,:)];
h_Nz_RStab_impulse_Clipped   = [h_Nz_RStab_impulse(1:500,:)];
h_Theta_RStab_impulse_Clipped = [h_Theta_RStab_impulse(1:500,:)];

%%%%%% Definition of response signals to be saved in GetData (*.gtd) format

Alpha = [h_Alpha_RStab_impulse_Clipped(1,1).*ones(100,1);
         h_Alpha_RStab_impulse_Clipped];

Q      = [h_Q_RStab_impulse_Clipped(1,1).*ones(100,1); h_Q_RStab_impulse_Clipped];
Nz     = [h_Nz_RStab_impulse_Clipped(1,1).*ones(100,1); h_Nz_RStab_impulse_Clipped];
Theta  = [h_Theta_RStab_impulse_Clipped(1,1).*ones(100,1);
         h_Theta_RStab_impulse_Clipped];

%%%%%% Creation and filtering of simulated ideal impulse input signal

Rstab  = [zeros(100,1); 67.*ones(1,1); zeros(length(h_Alpha_RStab_impulse_Clipped)-1,1)];

Rstab  = filtfilt(b,a,Rstab); % 9-pole low-pass filter with 20 rad/sec cutoff frequency

%%%%%% Definition of time and flight condition signals to be saved in GetData (*.gtd) format

Time   = (0:1/67:1/67*length(Alpha)-1/67);
Mach   = Mach(1:length(Alpha),1);
Qbar   = Qbar(1:length(Alpha),1);
Alt    = Alt(1:length(Alpha),1);
Vt     = Vt(1:length(Alpha),1);

%%%%%% Plot of above defined discrete pulse response and input signals

figure

subplot(5,1,1),plot(Time,Alpha,'k-');
title('Clipped Discrete Pulse Responses and Simulated Ideal Impulse Input vs Time (100

```

```

        'trim" points added equal to first point.),' FontName', 'Times', 'FontSize', 20);
ylabel('Alpha (deg)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20,'xticklabels',{''});

```

```

subplot(5,1,2),plot(Time,Q,'k-');
ylabel('Q (deg/sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20,'xticklabels',{''});

```

```

subplot(5,1,3),plot(Time,Nz,'k-');
ylabel('Nz (g)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20,'xticklabels',{''});

```

```

subplot(5,1,4),plot(Time,Theta,'k-');
ylabel('Theta (deg)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);

```

```

subplot(5,1,5),plot(Time,RStab,'k-');
ylabel('de (deg)', 'FontName', 'Times', 'FontSize', 20);
xlabel('Time (sec)', 'FontName', 'Times', 'FontSize', 20);
set(gca,'FontName','Times','FontSize',20);
input('');

```

%%%%%%%% Save discrete pulse response and simulated ideal impulse signals in GetData format

gdsave('C:\projects\Calm_SF3S04.gtd', Time, Alpha, Q, Nz, Theta, RStab, Mach, Qbar, Alt, Vt)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                 %
% The final section of HDPROCES first calculates and displays the average total fuel %
% weight for the current pass. This value is hand recorded by the user. HDPROCES then %
% calls the FORTRAN program MASSCALC.EXE to compute the test aircraft's total mass,%
% cg location, and moments and products of inertia. MASSCALC displays the answers %
% which are hand recorded by the user for PCPID input. The complete code for %
% MASSCALC is presented in the next section of this appendix. %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

%%%%%%%% Average fuel weight calculation

```

Avg_Wf = mean(input_file(:,actual_Wf))

```

%%%%%%%% Run MASSCALC program

```

!Masscalc

```

C.2 Mass Characteristics Program

The mass characteristics program, MASSCALC.EXE, was written by Mr. Chris Nagy of Quartic Engineering Inc. to compute the test aircraft's total mass, cg location, and moments and products of inertia given the aircraft's current total fuel weight. The MASSCALC calculations are for a Block 15 F-16B with a centerline fuel tank. The values given are based on a standard fuel burn (Nagy, 1997b).

The complete code for MASSCALC is presented in this section of Appendix C.

```
PROGRAM MASSCALC
```

```
IMPLICIT NONE
```

```
C
```

```
* This program computes the weight, center-of-gravity, and inertias for an F16-B model aircraft
* with a single centerline external tank. The fuel sequence, obtained from T.O 1F-16A-1, is as
* follows:
*   1) Centerline tank
*   2) Wing tanks equally
*   3) Aft and both forward tanks, delta shared equally between
*       aft and forward
*   4) Reservoir tanks
* Data for the various tanks and components were obtained from a General Dynamics memo on
* F-16 mass properties prepared by Mr. B.J. Aulds, Engineering Specialist and approved by
* Mr. E.A. Flores, Engineering Chief, Mass Properties (Nagy, 1997b). Data for the individual
* components are found in the "MP" array in this program.
*
```

```
INTEGER*2 I,J
```

```
REAL*4 WEIGHT, XCG, YCG, ZCG, IXX, IYY, IZZ, IXZ, OLD_WT
```

```
REAL*4 FUEL_WT, FUEL_LEFT, DELTA, XCGMAC, DX, DY, DZ
```

```
REAL*4 MP(24,8)
```

```
DATA ((MP(I,J), J=1,8), I=1,24)/
```

<u>*wt</u>	<u>sl</u>	<u>bl</u>	<u>wl</u>	<u>ix</u>	<u>iy</u>	<u>iz</u>	<u>ixz</u>	
- 17113.,	321.5,	0.0,	90.7,	6321.,	55026.5.,	54391.,	214.,	!B/b15/GU
- 56.,	244.9,	30.0,	98.0,	0.,	0.,	0.,	0.,	!EPU fuel

- 13.,	269.7,	9.5,	64.5,	0.,	0.,	0.,	0.,	!oxygen
- 13.,	315.0,	0.0,	100.0,	0.,	0.,	0.,	0.,	!fuel inert
- 16.,	128.8,	0.0,	92.3,	0.,	0.,	0.,	0.,	!surv kit
- 16.,	193.8,	0.0,	94.8,	0.,	0.,	0.,	0.,	!surv kit
- 190.,	137.7,	0.0,	105.5,	0.,	0.,	0.,	0.,	!pilot(mod)
- 190.,	202.8,	0.0,	108.1,	0.,	0.,	0.,	0.,	!pilot(mod)
- 55.,	426.4,	0.0,	90.0,	0.,	0.,	0.,	0.,	!resid fluid
- 24.,	375.0,	0.0,	80.0,	0.,	0.,	0.,	0.,	!engine oil
- 740.,	230.9,	0.0,	104.4,	0.,	0.,	0.,	0.,	!F1 fuel
- 684.,	297.0,	0.0,	101.9,	87.,	8.,	80.,	0.,	!F2 fuel
- 442.,	326.0,	-25.0,	97.8,	0.,	0.,	0.,	0.,	!LRes fuel
- 442.,	326.0,	25.0,	97.8,	0.,	0.,	0.,	0.,	!RRes fuel
- 2313.,	389.5,	0.0,	99.5,	316.,	354.,	459.,	0.,	!Aft fuel
- 1143.,	350.0,	0.0,	92.0,	1961.,	135.,	2096.,	0.,	!Wing fuel
- 22.,	370.0,	0.0,	84.6,	0.,	0.,	0.,	0.,	!Line fuel
- 1950.,	319.6,	0.0,	38.5,	7.,	593.,	612.,	0.,	!300ECL fuel
- 12.,	290.0,	0.0,	25.6,	0.,	0.,	0.,	0.,	!300ECL unfuel
- 69.,	366.3,	-182.6,	91.6,	0.,	11.,	11.,	0.,	!L 16S210
- 69.,	366.3,	182.6,	91.6,	0.,	11.,	11.,	0.,	!R 16S210
- 172.,	307.2,	0.0,	52.5,	0.,	17.,	17.,	0.,	!Univ pylon
- 371.,	327.0,	0.0,	38.1,	10.,	139.,	139.,	1.,	!300G tank
- 24.,	457.0,	35.7,	90.5,	0.,	0.,	0.,	0./	!Chaff

C---Start with Basic Airframe (Fuel-less) Mass Properties

```

WEIGHT = MP(1,1)
XCG     = MP(1,2)
YCG     = MP(1,3)
ZCG     = MP(1,4)
IXX     = MP(1,5)
IYY     = MP(1,6)
IZZ     = MP(1,7)
IXZ     = MP(1,8)

```

C---Compute Fuel Locations and Proportion Component Inertias for Partly-full Fuel Tanks

```

WRITE(6, '(1H0, A, $)') 'Enter Fuel Weight - '
READ(5, '(F10.0)') FUEL_WT

```

! Start with fuel in lines and residue in external tank

FUEL_LEFT = FUEL_WT - MP(17,1) - MP(19,1)

! Add-In Reservoir Tanks (Items 13,14)

IF (FUEL_LEFT .LT. MP(13,1)+MP(14,1)) THEN

MP(11,1) = 0. !F1 Tank
MP(12,1) = 0. !F2 Tank
MP(12,5) = 0.
MP(12,6) = 0.
MP(12,7) = 0.
MP(13,1) = FUEL_LEFT / 2. !Left Reservoir
MP(14,1) = FUEL_LEFT / 2. !Right Reservoir
MP(15,1) = 0. !Aft Tank
MP(15,5) = 0.
MP(15,6) = 0.
MP(15,7) = 0.
MP(16,1) = 0. !Wing Tanks
MP(16,5) = 0.
MP(16,6) = 0.
MP(16,7) = 0.
MP(18,1) = 0. !External Tank
MP(18,5) = 0.
MP(18,6) = 0.
MP(18,7) = 0.
FUEL_LEFT = 0.0
GO TO 10

ELSE

FUEL_LEFT = FUEL_LEFT - MP(13,1) - MP(14,1)

ENDIF

! Add-In Remainder of Aft Tank (Item 15)

DELTA = MP(15,1) - MP(11,1) - MP(12,1)

IF (FUEL_LEFT .LT. DELTA) THEN

MP(11,1) = 0. !F1 Tank
MP(12,1) = 0. !F2 Tank
MP(12,5) = 0.
MP(12,6) = 0.

```

MP(12,7)    = 0.
MP(15,5)    = MP(15,5) * FUEL_LEFT / MP(15,1)    !Aft Tank
MP(15,6)    = MP(15,6) * FUEL_LEFT / MP(15,1)
MP(15,7)    = MP(15,7) * FUEL_LEFT / MP(15,1)
MP(15,1)    = FUEL_LEFT
MP(16,1)    = 0.                                !Wing Tanks
MP(16,5)    = 0.
MP(16,6)    = 0.
MP(16,7)    = 0.
MP(18,1)    = 0.                                !External Tank
MP(18,5)    = 0.
MP(18,6)    = 0.
MP(18,7)    = 0.
FUEL_LEFT = 0.0
GO TO 10

ELSE

    FUEL_LEFT = FUEL_LEFT - DELTA

ENDIF

!   Add-In Balance of Forward and Aft Fuselage Tanks (Items 11,12,15)

IF (FUEL_LEFT .LT. MP(11,1)+MP(12,1)+MP(15,1)-DELTA) THEN

    IF (FUEL_LEFT/2 .LT. MP(12,1)) THEN

        MP(11,1) = 0.                            !F1 Tank
        MP(12,5) = MP(12,5) * (FUEL_LEFT/2) / MP(12,1)    !F2 Tank
        MP(12,6) = MP(12,6) * (FUEL_LEFT/2) / MP(12,1)
        MP(12,7) = MP(12,7) * (FUEL_LEFT/2) / MP(12,1)
        MP(12,1) = (FUEL_LEFT/2)

    ELSE

        MP(11,1) = (FUEL_LEFT/2) - MP(12,1)            !F1 Tank

    ENDIF

    MP(15,5) = MP(15,5) * (FUEL_LEFT/2+DELTA) / MP(15,1)    !Aft Tank
    MP(15,6) = MP(15,6) * (FUEL_LEFT/2+DELTA) / MP(15,1)
    MP(15,7) = MP(15,7) * (FUEL_LEFT/2+DELTA) / MP(15,1)
    MP(15,1) = (FUEL_LEFT/2+DELTA)
    MP(16,1) = 0.                                !Wing Tanks
    MP(16,5) = 0.

```

```

MP(16,6)    = 0.
MP(16,7)    = 0.
MP(18,1)    = 0.                                !External Tank
MP(18,5)    = 0.
MP(18,6)    = 0.
MP(18,7)    = 0.
FUEL_LEFT = 0.0
GO TO 10

```

ELSE

```

FUEL_LEFT = FUEL_LEFT - (MP(11,1)+MP(12,1)+MP(15,1)-DELTA)

```

ENDIF

! Add-In Wing Tanks (Item 16)

```

IF (FUEL_LEFT .LT. MP(16,1)) THEN
  MP(16,5)    = MP(16,5) * FUEL_LEFT / MP(16,1)    !Wing Tanks
  MP(16,6)    = MP(16,6) * FUEL_LEFT / MP(16,1)
  MP(16,7)    = MP(16,7) * FUEL_LEFT / MP(16,1)
  MP(16,1)    = FUEL_LEFT
  MP(18,1)    = 0.                                !External Tank
  MP(18,5)    = 0.
  MP(18,6)    = 0.
  MP(18,7)    = 0.
  FUEL_LEFT = 0.0
  GO TO 10

```

ELSE

```

FUEL_LEFT = FUEL_LEFT - MP(16,1)

```

ENDIF

! Add-In Centerline External Tank (Item 18)

```

IF (FUEL_LEFT .LT. MP(18,1)) THEN
  MP(18,5)    = MP(18,5) * FUEL_LEFT / MP(18,1)    !External Tank
  MP(18,6)    = MP(18,6) * FUEL_LEFT / MP(18,1)
  MP(18,7)    = MP(18,7) * FUEL_LEFT / MP(18,1)
  MP(18,1)    = FUEL_LEFT
  FUEL_LEFT = 0.0
  GO TO 10

```

ENDIF

C---Compute Weight and CGs

! Basic Equipment (Items 2-10)

10 DO I = 2,10

OLD_WT = WEIGHT
WEIGHT = WEIGHT + MP(I,1)
XCG = (OLD_WT * XCG + MP(I,1) * MP(I,2)) / WEIGHT
YCG = (OLD_WT * YCG + MP(I,1) * MP(I,3)) / WEIGHT
ZCG = (OLD_WT * ZCG + MP(I,1) * MP(I,4)) / WEIGHT

ENDDO

! Fuel Components (Items 11-19)

DO I = 11,19

OLD_WT = WEIGHT
WEIGHT = WEIGHT + MP(I,1)
XCG = (OLD_WT * XCG + MP(I,1) * MP(I,2)) / WEIGHT
YCG = (OLD_WT * YCG + MP(I,1) * MP(I,3)) / WEIGHT
ZCG = (OLD_WT * ZCG + MP(I,1) * MP(I,4)) / WEIGHT

ENDDO

! Stores (Empty External Tank, AIM-9 Launchers) (Items 20-24)

DO I = 20,24

OLD_WT = WEIGHT
WEIGHT = WEIGHT + MP(I,1)
XCG = (OLD_WT * XCG + MP(I,1) * MP(I,2)) / WEIGHT
YCG = (OLD_WT * YCG + MP(I,1) * MP(I,3)) / WEIGHT
ZCG = (OLD_WT * ZCG + MP(I,1) * MP(I,4)) / WEIGHT

ENDDO

C---Compute Inertias

! Basic Equipment (Items 2-10)

DO I = 2,10

```
IXX = IXX + MP(I,5) + MP(I,1) / 32.17405 *  
- ((MP(I,3) - YCG)**2 + (MP(I,4) - ZCG)**2) / 144  
IYY = IYY + MP(I,6) + MP(I,1) / 32.17405 *  
- ((MP(I,2) - XCG)**2 + (MP(I,4) - ZCG)**2) / 144  
IZZ = IZZ + MP(I,7) + MP(I,1) / 32.17405 *  
- ((MP(I,2) - XCG)**2 + (MP(I,3) - YCG)**2) / 144  
IXZ = IXZ + MP(I,8) + MP(I,1) / 32.17405 *  
- ((XCG - MP(I,2)) * (ZCG - MP(I,4))) / 144
```

ENDDO

! Fuel Components (Items 11-19)

DO I = 11,19

```
IXX = IXX + MP(I,5) + MP(I,1) / 32.17405 *  
- ((MP(I,3) - YCG)**2 + (MP(I,4) - ZCG)**2) / 144  
IYY = IYY + MP(I,6) + MP(I,1) / 32.17405 *  
- ((MP(I,2) - XCG)**2 + (MP(I,4) - ZCG)**2) / 144  
IZZ = IZZ + MP(I,7) + MP(I,1) / 32.17405 *  
- ((MP(I,2) - XCG)**2 + (MP(I,3) - YCG)**2) / 144  
IXZ = IXZ + MP(I,8) + MP(I,1) / 32.17405 *  
- ((XCG - MP(I,2)) * (ZCG - MP(I,4))) / 144
```

ENDDO

! Add In Effects of Stores (Empty External Tank, AIM-9 Launchers, chaff)

DO I = 20,24

```
IXX = IXX + MP(I,5) + MP(I,1) / 32.17405 *  
- ((MP(I,3) - YCG)**2 + (MP(I,4) - ZCG)**2) / 144  
IYY = IYY + MP(I,6) + MP(I,1) / 32.17405 *  
- ((MP(I,2) - XCG)**2 + (MP(I,4) - ZCG)**2) / 144  
IZZ = IZZ + MP(I,7) + MP(I,1) / 32.17405 *  
- ((MP(I,2) - XCG)**2 + (MP(I,3) - YCG)**2) / 144  
IXZ = IXZ + MP(I,8) + MP(I,1) / 32.17405 *  
- ((XCG - MP(I,2)) * (ZCG - MP(I,4))) / 144
```

ENDDO

C---Compute MAC CG and Delta CGs in Feet

XCGMAC = (XCG - 273.11) / 135.84 * 100.
DX = (XCG - 320.65) / 12
DY = (YCG - 0.) / 12.
DZ = (ZCG - 91.) / 12

C---Output Mass Properties

WRITE(6, '(1H0, A, 2X, 3(1X,A,3X), 4(2X,A,3X))')
- 'Weight', 'X-cg(%MAC)', 'Y-cg', 'Z-cg', 'Ixx', 'Iyy', 'Izz', 'Ixz'

WRITE(6, '(1H0, F6.0, 2X, F6.2, 1H(, F4.1, 1H),
- 2(F6.2,2X), 1X, 4(F6.0,2X))')
- WEIGHT, XCG, XCGMAC, YCG, ZCG, IXX, IYY, IZZ, IXZ

WRITE(6, '(1H , 8X, 3(F6.2,2X))')
- DX, DY, DZ

STOP
END

Appendix D : Calm Air Stability Derivative Estimation Results

This appendix contains detailed calm air stability derivative estimation results. The definitions in Table D.1 were used to categorize the results from different passes.

Table D.1. Turbulence Level Definitions

Turbulence Level	Δ Normal Acceleration (Δg) ¹
Calm Air	$ \Delta n_z \leq 0.1$
Turbulent Air	$0.3 \leq \Delta n_z $

¹Change in normal acceleration from 1-g at a frequency of 1 radian per second

All passes from Flights 1 through 3 were in calm air according to these definitions. Actual turbulence levels from Flights 1 through 3 are presented in Table D.2.

Table D.2 also presents the calm air stability derivative estimate values and associated Cramer-Rao bounds for $C_{N\alpha}$, $C_{N\delta e}$, $C_{m\alpha}$, C_{mq} and $C_{m\delta e}$. Results for all 32 calm air data collection passes were calculated from doublet input and response data using PCPID alone. The Cramer-Rao bound was a measure of the confidence PCPID had in each stability derivative estimate. The smaller the Cramer-Rao bound, the more confidence PCPID had in the estimate. The Cramer-Rao bound values presented in Table D.2 were the actual bound multiplied by a factor of 10. This factor has been commonly used at the AFFTC to account for the fact that the true noise present on the signals processed by PCPID is correlated, not uncorrelated as assumed by the parameter estimation routine in PCPID, and to more closely correlate the size of the bounds to the scatter of the estimates (Maine and Iliff, 1986:135 to 137).

Time history match cost function values for each of the calm air passes are given in Table D.3. The cost function values were a quantitative indication of how closely the PCPID computed time histories matched the measured time histories. The derivative estimates for a run with smaller cost function values were considered more accurate than those estimates associated with larger cost function values. The values of the cost functions were equalized at unity for the first 10 calm air passes by varying response parameter weightings within PCPID. These weightings, which were used for all subsequent PCPID processing, are given at the top of Table D.3.

The stability derivative estimate mean values, associated confidence intervals based on a 95 percent confidence level, and the data set standard deviations are presented at the top of Tables D.2 and D.3. The 95% confidence level intervals were calculated using Equation (D.1).

$$\mu = \bar{x} \pm t_{v, 1 - \frac{\alpha}{2}} \left(\frac{s}{\sqrt{n}} \right) \quad (D.1)$$

where

μ = population mean

\bar{x} = sample mean

$t_{v, 1 - \frac{\alpha}{2}}$ = student's t distribution statistic

v = degrees of freedom ($n - 1$)

α = uncertainty level (0.05 for 95% confidence level)

s = sample standard deviation

n = number of samples

Table D.2. Calm Air Stability Derivative Estimate Values and Cramer-Rao Bounds

						$C_{N\alpha}$	CRbd ⁷	$C_{N\delta e}$	CRbd	$C_{m\alpha}$	CRbd	C_{mq}	CRbd	$C_{m\delta e}$	CRbd
Mean ⁴						0.0714	4.38E-03	0.0132	4.76E-03	0.0008	5.70E-04	-3.1380	1.10E+00	-0.0114	9.90E-04
Mean Confidence Interval ⁵						0.0009	6.63E-04	0.0007	5.21E-04	0.0005	1.51E-04	0.5339	2.22E-01	0.0008	2.07E-04
Desired Confidence Interval						2.5E-03		2.0E-03		2.5E-04		4.0E-01		4.0E-04	
Confidence Level ⁶						1.0000		1.0000		0.6606		0.8634		0.7056	
Standard Deviation						0.0025	1.84E-03	0.0019	1.45E-03	0.0015	4.20E-04	1.4808	6.15E-01	0.0021	5.74E-04
Flight	Pass	Δg ¹	Xcg ² (pt)	Avg α ³		$C_{N\alpha}$	CRbd	$C_{N\delta e}$	CRbd	$C_{m\alpha}$	CRbd	C_{mq}	CRbd	$C_{m\delta e}$	CRbd
1	5	0.07	35.60	2.11		0.0708	5.24E-03	0.0133	4.35E-03	0.0000	5.77E-04	-4.4123	1.43E+00	-0.0103	9.46E-04
1	6	0.01	35.13	2.07		0.0681	4.27E-03	0.0153	3.77E-03	0.0004	3.42E-04	-5.5375	1.18E+00	-0.0110	7.63E-04
1	7	0.08	35.72	1.98		0.0729	4.14E-03	0.0132	3.65E-03	-0.0002	9.09E-04	-4.6108	1.92E+00	-0.0105	1.14E-03
1	8	0.09	35.82	2.00		0.0623	4.76E-03	0.0152	4.92E-03	0.0001	5.64E-04	-4.5818	1.77E+00	-0.0106	1.09E-03
2	1	0.04	35.33	2.38		0.0686	5.52E-03	0.0141	5.03E-03	0.0013	3.87E-04	-1.4549	7.87E-01	-0.0114	8.26E-04
2	2	0.02	35.16	2.23		0.0715	3.70E-03	0.0126	3.83E-03	0.0025	6.11E-04	-4.7675	1.13E+00	-0.0134	9.82E-04
2	3	0.02	35.14	1.94		0.0732	5.08E-03	0.0131	5.08E-03	0.0045	9.48E-04	-5.7164	1.59E+00	-0.0151	1.39E-03
2	4	0.01	35.08	2.13		0.0710	4.08E-03	0.0129	4.30E-03	0.0028	6.87E-04	-4.5500	1.28E+00	-0.0138	1.08E-03
2	5	0.03	35.28	2.09		0.0700	4.60E-03	0.0124	5.27E-03	0.0020	5.52E-04	-2.3454	9.61E-01	-0.0131	1.01E-03
2	6	0.03	35.28	1.98		0.0729	5.33E-03	0.0116	5.03E-03	0.0025	7.67E-04	-3.3342	1.27E+00	-0.0133	1.23E-03
2	7	0.06	35.54	1.84		0.0753	6.84E-03	0.0094	7.04E-03	0.0031	1.40E-03	-4.0800	2.13E+00	-0.0139	1.99E-03
2	8	0.05	35.44	1.89		0.0728	5.93E-03	0.0122	6.16E-03	0.0018	9.42E-04	-2.8480	1.53E+00	-0.0129	1.54E-03
2	9	0.02	35.20	1.87		0.0672	1.16E-02	0.0154	1.07E-02	-0.0008	1.76E-03	-1.1682	2.30E+00	-0.0094	2.86E-03
2	10	0.03	35.24	1.83		0.0744	4.77E-03	0.0107	4.12E-03	0.0020	8.96E-04	-4.0223	1.46E+00	-0.0135	1.23E-03
2	11	0.04	35.33	1.92		0.0743	4.56E-03	0.0093	6.23E-03	0.0021	8.75E-04	-4.7754	1.69E+00	-0.0150	1.61E-03
2	12	0.03	35.22	1.93		0.0717	5.90E-03	0.0123	5.32E-03	0.0003	8.29E-04	-1.8626	1.30E+00	-0.0117	1.33E-03
2	13	0.01	35.10	2.00		0.0715	4.71E-03	0.0137	4.46E-03	0.0003	6.91E-04	-2.8059	1.21E+00	-0.0122	1.22E-03
2	14	0.04	35.32	1.76		0.0741	5.76E-03	0.0104	5.63E-03	0.0012	7.31E-04	-2.6277	1.29E+00	-0.0129	1.32E-03
2	15	0.05	35.46	1.70		0.0717	5.60E-03	0.0140	5.66E-03	0.0023	1.22E-03	-5.0852	2.31E+00	-0.0147	1.80E-03
3	1	0.02	35.20	2.53		0.0696	4.28E-03	0.0137	4.38E-03	0.0026	6.97E-04	-4.8283	1.36E+00	-0.0134	1.07E-03
3	2	0.07	35.66	2.11		0.0710	2.96E-03	0.0149	3.92E-03	-0.0001	1.61E-04	-2.3739	4.17E-01	-0.0095	4.23E-04
3	4	0.04	35.39	2.10		0.0701	2.95E-03	0.0153	3.73E-03	0.0001	2.10E-04	-3.6422	5.57E-01	-0.0098	5.23E-04
3	5	0.04	35.36	1.98		0.0698	2.36E-03	0.0156	2.99E-03	-0.0006	1.42E-04	-1.8586	3.71E-01	-0.0092	3.55E-04
3	6	0.03	35.26	1.86		0.0729	3.36E-03	0.0119	3.82E-03	-0.0010	1.94E-04	-1.4672	4.48E-01	-0.0092	4.51E-04
3	7	0.04	35.31	2.02		0.0706	2.49E-03	0.0154	3.32E-03	-0.0005	1.45E-04	-2.4108	4.04E-01	-0.0096	3.92E-04
3	8	0.06	35.52	1.72		0.0716	3.41E-03	0.0126	3.92E-03	-0.0009	1.86E-04	-1.6525	4.24E-01	-0.0096	4.56E-04
3	9	0.02	35.15	1.76		0.0730	3.60E-03	0.0105	4.25E-03	-0.0012	1.51E-04	-0.7620	4.11E-01	-0.0088	4.27E-04
3	11	0.08	35.68	1.67		0.0721	2.12E-03	0.0127	3.15E-03	-0.0006	1.08E-04	-1.9607	3.22E-01	-0.0092	3.42E-04
3	12	0.02	35.17	1.69		0.0712	2.99E-03	0.0156	5.87E-03	-0.0008	1.11E-04	-0.3922	3.76E-01	-0.0079	3.99E-04
3	13	0.04	35.36	1.64		0.0714	2.38E-03	0.0156	3.92E-03	-0.0002	1.44E-04	-2.0307	4.60E-01	-0.0090	4.22E-04
3	14	0.05	35.46	1.47		0.0733	2.52E-03	0.0129	4.16E-03	0.0000	1.40E-04	-2.6051	5.02E-01	-0.0089	4.75E-04
3	15	0.07	35.62	1.51		0.0729	2.19E-03	0.0150	4.43E-03	0.0003	1.66E-04	-3.8444	5.78E-01	-0.0105	5.99E-04

¹Change in normal acceleration from 1-g at a frequency of 1 radian per second²Xcg location measured in percent of mean aerodynamic chord³Average angle of attack for pass⁴Mean based on 32 samples⁵Confidence interval of the mean based on a 95 percent confidence level⁶Confidence level based on Desired Confidence Interval⁷Cramer-Rao bounds have been multiplied by a factor of 10

Table D.3. Calm Air Time History Match Cost Function Values

		Cost α ⁴	Cost q ⁵	Cost θ ⁶	Cost n_z ⁷
Mean ¹		3.24	2.11	1.85	3.48
Mean Confidence Interval ²		1.32	0.54	0.48	1.57
Standard Deviation		3.66	1.50	1.32	4.35
Parameter Weighting ³		23	3	14	60
Flight	Pass	Cost α	Cost q	Cost θ	Cost n_z
1	5	8.33	1.53	2.01	6.26
1	6	2.54	1.63	0.71	2.55
1	7	3.67	1.30	0.78	3.56
1	8	5.73	2.94	1.36	8.07
2	1	3.06	2.28	2.47	2.87
2	2	1.16	1.66	1.31	1.41
2	3	1.47	3.20	2.16	0.97
2	4	1.87	0.40	1.51	1.23
2	5	1.51	1.84	1.87	0.63
2	6	1.68	2.37	1.45	1.15
2	7	3.09	4.30	1.86	3.02
2	8	1.95	4.02	1.58	1.69
2	9	21.60	8.60	6.86	24.60
2	10	1.29	1.17	1.20	1.00
2	11	1.98	1.11	1.36	1.60
2	12	2.44	2.70	1.94	2.72
2	13	1.79	1.16	0.80	1.79
2	14	3.06	2.99	1.77	2.02
2	15	3.74	0.60	1.67	2.00
3	1	2.19	1.23	2.51	1.85
3	2	2.08	1.57	1.00	3.67
3	4	2.70	2.24	0.52	2.24
3	5	1.19	1.81	1.69	1.53
3	6	2.47	1.67	2.68	4.27
3	7	1.78	1.58	1.08	1.45
3	8	3.33	1.66	4.40	4.89
3	9	4.01	1.69	3.71	6.75
3	11	1.37	1.19	0.76	1.82
3	12	4.86	3.23	3.95	8.25
3	13	1.83	1.26	0.84	1.72
3	14	2.15	1.80	0.76	2.46
3	15	1.90	0.69	0.67	1.45

¹Mean based on 32 samples²Confidence interval of the mean based on a 95 percent confidence level³Mean parameter weighting from first ten calm air PCPID runs⁴Cost function value associated with matching angle of attack measured time history⁵Cost function value associated with matching pitch rate measured time history⁶Cost function value associated with matching pitch angle measured time history⁷Cost function value associated with matching normal acceleration measured time history

Appendix E : Turbulent Air Stability Derivative Estimation Results

This appendix contains detailed turbulent air stability derivative estimation results. The definitions in Table E.1 were used to categorize the results from different passes.

Table E.1. Turbulence Level Definitions

Turbulence Level	Δ Normal Acceleration (Δg) ¹
Calm Air	$ \Delta n_z \leq 0.1$
Turbulent Air	$0.3 \leq \Delta n_z $

¹Change in normal acceleration from 1-g at a frequency of 1 radian per second

Fifty-two passes from Flights 4 through 10 were in turbulent air according to these definitions. Actual turbulence levels from Flights 4 through 10 are presented in Table E.2.

Table E.2 also presents the turbulent air stability derivative estimate values and associated Cramer-Rao bounds for $C_{N\alpha}$, $C_{N\delta_e}$, $C_{m\alpha}$, C_{mq} and $C_{m\delta_e}$. Results for all 52 turbulent air data collection passes were calculated from doublet input and response data using PCPID alone. The Cramer-Rao bound was a measure of the confidence PCPID had in each stability derivative estimate. The smaller the Cramer-Rao bound, the more confidence PCPID had in the estimate. The Cramer-Rao bound values presented in Table E.2 were the actual bound multiplied by a factor of 10. This factor has been commonly used at the AFFTC to account for the fact that the true noise present on the signals processed by PCPID is correlated, not uncorrelated as assumed by the parameter estimation routine in PCPID, and to more closely correlate the size of the bounds to the scatter of the estimates (Maine and Iliff, 1986:135 to 137).

Time history match cost function values for each of the turbulent air passes are given in Table E.3. The cost function values were a quantitative indication of how closely the PCPID computed time histories matched the measured time histories. The derivative estimates for a run with smaller cost function values were considered more accurate than those estimates associated with larger cost function values. The values of the cost functions were equalized at unity for the first 10 calm air passes by varying response parameter weightings within PCPID. These weightings, used for all subsequent PCPID processing, are given at the top of Table E.3.

The stability derivative estimate mean values, associated confidence intervals based on a 95 percent confidence level, and the data set standard deviations are presented at the top of Tables E.2 and E.3. The 95% confidence level intervals were calculated using Equation (E.1).

$$\mu = \bar{x} \pm t_{v, 1 - \frac{\alpha}{2}} \left(\frac{s}{\sqrt{n}} \right) \quad (E.1)$$

where

μ = population mean

\bar{x} = sample mean

$t_{v, 1 - \frac{\alpha}{2}}$ = student's t distribution statistic

v = degrees of freedom ($n - 1$)

α = uncertainty level (0.05 for 95% confidence level)

s = sample standard deviation

n = number of samples

Table E.2. Turbulent Air Stability Derivative Estimate Values and Cramer-Rao Bounds

					$C_{N\alpha}$ CR bnd		$C_{N\dot{\alpha}}$ CR bnd		$C_{m\alpha}$ CR bnd		C_{mq} CR bnd		$C_{m\dot{\alpha}}$ CR bnd	
Mean ⁴					0.0693	6.49E-03	0.0174	8.63E-03	0.0010	8.03E-04	-3.5564	1.89E+00	-0.0111	1.61E-03
Mean Confidence Interval ⁵					1.67E-03	5.89E-04	2.26E-03	9.06E-04	3.99E-04	1.49E-04	7.91E-01	3.13E-01	7.80E-04	2.20E-04
Desired Confidence Interval					2.50E-03		2.00E-03		2.50E-04		4.00E-01		4.00E-04	
Confidence Level ⁶					99.6%		91.9%		78.6%		68.5%		69.2%	
Standard Deviation					0.0060	2.12E-03	0.0081	3.25E-03	0.0014	5.34E-04	2.8403	1.12E+00	0.0028	7.90E-04
Flight	Pass	$\Delta\alpha^1$	X-cg ² (pct)	Avg α^3	$C_{N\alpha}$	CR bnd	$C_{N\dot{\alpha}}$	CR bnd	$C_{m\alpha}$	CR bnd	C_{mq}	CR bnd	$C_{m\dot{\alpha}}$	CR bnd
4	4	0.36	38.22	1.91	0.0709	7.54E-03	0.0084	8.18E-03	0.0021	1.36E-03	-4.3194	2.66E+00	-0.0116	2.05E-03
4	6	0.37	38.25	1.97	0.0628	6.26E-03	0.0217	8.51E-03	-0.0013	1.67E-04	1.6780	5.49E-01	-0.0066	5.81E-04
4	7	0.35	38.13	1.62	0.0697	9.24E-03	0.0088	9.03E-03	0.0014	1.94E-03	-1.3382	2.66E+00	-0.0117	2.56E-03
4	8	0.79	42.01	1.89	0.0714	5.57E-03	0.0117	4.46E-03	0.0020	8.69E-04	-3.9896	1.59E+00	-0.0125	1.16E-03
4	10	0.56	39.98	1.70	0.0828	8.95E-03	0.0046	7.21E-03	0.0008	1.11E-03	-3.8905	2.22E+00	-0.0122	1.71E-03
4	16	0.38	38.36	1.67	0.0749	4.78E-03	0.0114	4.88E-03	0.0024	8.83E-04	-6.3715	1.80E+00	-0.0148	1.46E-03
4	17	0.34	38.01	1.60	0.0752	5.74E-03	0.0100	5.78E-03	0.0021	9.29E-04	-5.2690	1.71E+00	-0.0146	1.54E-03
4	18	0.33	37.94	1.38	0.0756	6.01E-03	0.0097	4.93E-03	0.0024	7.83E-04	-5.3536	1.52E+00	-0.0145	1.34E-03
5	1	0.65	40.76	2.17	0.0723	4.40E-03	0.0063	6.94E-03	-0.0002	3.50E-04	-1.3737	7.18E-01	-0.0089	8.95E-04
5	2	0.34	38.02	1.94	0.0705	7.77E-03	0.0168	1.28E-02	-0.0009	2.62E-04	0.8676	7.35E-01	-0.0070	8.05E-04
5	3	0.33	37.91	2.28	0.0616	5.88E-03	0.0259	9.65E-03	0.0010	5.97E-04	-3.8764	1.50E+00	-0.0102	1.51E-03
5	4	0.45	38.93	1.86	0.0690	5.83E-03	0.0237	9.36E-03	0.0004	3.75E-04	-1.9526	1.08E+00	-0.0096	1.20E-03
5	5	0.37	38.31	2.08	0.0704	3.89E-03	0.0179	5.42E-03	0.0008	2.98E-04	-4.1945	8.15E-01	-0.0109	8.33E-04
5	6	0.54	39.77	1.94	0.0632	6.46E-03	0.0209	8.83E-03	-0.0011	2.13E-04	1.5411	5.77E-01	-0.0071	7.12E-04
5	7	0.36	38.20	1.96	0.0647	5.92E-03	0.0179	8.14E-03	-0.0011	1.87E-04	1.1611	5.60E-01	-0.0072	6.41E-04
5	9	0.35	38.12	1.52	0.0654	4.90E-03	0.0215	7.32E-03	-0.0005	2.33E-04	-1.3774	7.76E-01	-0.0074	7.51E-04
5	11	0.50	39.38	1.74	0.0678	5.16E-03	0.0116	6.69E-03	-0.0006	2.12E-04	-2.6699	8.36E-01	-0.0080	7.00E-04
5	12	0.62	40.49	1.80	0.0690	4.52E-03	0.0192	5.83E-03	-0.0008	2.70E-04	-1.0655	6.37E-01	-0.0091	7.79E-04
5	13	0.52	39.57	1.52	0.0722	5.80E-03	0.0167	1.01E-02	-0.0005	1.90E-04	0.3229	6.58E-01	-0.0082	7.71E-04
6	6	0.38	38.33	2.22	0.0735	3.60E-03	0.0103	6.78E-03	0.0013	1.43E-03	-4.7636	2.84E+00	-0.0134	2.71E-03
6	7	0.31	37.77	2.42	0.0704	5.44E-03	0.0184	9.85E-03	0.0015	5.37E-04	-3.2032	1.23E+00	-0.0139	1.55E-03
6	8	0.30	37.67	1.58	0.0735	3.76E-03	0.0157	6.32E-03	0.0031	6.10E-04	-8.1452	1.55E+00	-0.0161	1.42E-03
6	10	0.32	37.82	1.98	0.0731	5.03E-03	0.0110	8.26E-03	0.0017	9.14E-04	-4.8711	1.90E+00	-0.0143	2.11E-03
6	12	0.42	38.68	1.47	0.0696	4.87E-03	0.0170	9.77E-03	0.0022	7.64E-04	-6.8179	2.23E+00	-0.0140	1.88E-03
6	14	0.64	40.66	1.43	0.0713	4.59E-03	0.0143	5.81E-03	0.0035	1.40E-03	-7.0112	2.45E+00	-0.0156	2.13E-03
8	6	0.37	38.26	2.04	0.0724	4.68E-03	0.0107	4.86E-03	0.0035	8.43E-04	-7.4791	1.41E+00	-0.0145	1.29E-03
8	10	0.35	38.11	1.92	0.0670	3.51E-03	0.0175	3.93E-03	0.0027	5.54E-04	-5.0190	9.46E-01	-0.0137	9.20E-04
8	11	0.57	40.01	2.06	0.0698	5.77E-03	0.0154	6.97E-03	0.0028	9.07E-04	-4.5197	1.44E+00	-0.0143	1.60E-03
8	14	0.37	38.26	1.77	0.0683	6.24E-03	0.0155	6.45E-03	0.0022	1.10E-03	-7.0042	2.33E+00	-0.0132	1.61E-03
8	17	0.60	40.31	1.50	0.0754	5.63E-03	0.0107	5.69E-03	0.0011	7.99E-04	-4.6725	1.63E+00	-0.0120	1.31E-03
8	19	0.45	38.93	1.85	0.0473	1.04E-02	0.0432	1.12E-02	0.0041	2.41E-03	-2.1807	2.51E+00	-0.0165	3.49E-03
8	20	0.36	38.20	1.65	0.0722	5.97E-03	0.0133	6.51E-03	0.0016	8.41E-04	-4.1524	1.43E+00	-0.0131	1.46E-03
8	21	0.53	39.65	1.40	0.0665	1.01E-02	0.0112	9.94E-03	0.0024	1.51E-03	-6.8145	3.20E+00	-0.0114	2.46E-03
8	23	0.59	40.20	1.35	0.0783	6.43E-03	0.0078	5.73E-03	0.0016	7.45E-04	-4.8968	1.38E+00	-0.0122	1.29E-03
9	1	0.49	39.36	1.95	0.0677	6.99E-03	0.0250	1.04E-02	0.0037	1.75E-03	-7.5557	3.89E+00	-0.0147	3.10E-03
9	3	0.33	37.91	2.13	0.0695	3.55E-03	0.0179	4.15E-03	0.0003	3.58E-04	-4.0784	8.71E-01	-0.0106	8.26E-04
9	4	0.82	42.24	2.59	0.0670	5.69E-03	0.0189	7.67E-03	0.0005	6.28E-04	-4.7841	1.81E+00	-0.0109	1.44E-03
9	5	0.36	38.20	2.10	0.0851	7.81E-03	0.0097	8.21E-03	0.0013	9.77E-04	-7.8236	2.76E+00	-0.0125	2.14E-03
9	7	0.50	39.42	2.02	0.0593	9.43E-03	0.0252	1.53E-02	0.0002	5.28E-04	-3.4505	2.40E+00	-0.0111	1.90E-03
9	8	0.34	38.01	2.00	0.0686	7.04E-03	0.0158	8.78E-03	-0.0010	6.96E-04	0.2452	1.72E+00	-0.0090	1.77E-03
9	9	0.79	41.95	1.79	0.0753	1.07E-02	0.0131	1.14E-02	-0.0007	2.03E-03	-2.7086	3.99E+00	-0.0089	3.33E-03
9	10	0.70	41.17	1.89	0.0619	8.25E-03	0.0363	1.42E-02	0.0009	1.82E-03	-4.5366	5.49E+00	-0.0111	4.01E-03
9	11	0.40	38.54	1.57	0.0730	7.42E-03	0.0074	1.08E-02	-0.0005	6.96E-04	-1.1400	1.84E+00	-0.0080	1.74E-03
9	12	0.44	38.92	1.59	0.0659	6.24E-03	0.0184	7.80E-03	0.0000	6.78E-04	-2.9718	2.21E+00	-0.0091	1.52E-03
9	13	0.30	37.68	1.72	0.0736	1.07E-02	0.0096	9.96E-03	0.0020	1.56E-03	-11.0785	5.80E+00	-0.0112	3.27E-03
9	15	0.33	37.91	1.75	0.0600	5.41E-03	0.0326	8.38E-03	0.0005	5.03E-04	-4.4206	1.77E+00	-0.0104	1.60E-03
9	16	0.68	40.99	1.69	0.0620	6.71E-03	0.0310	1.02E-02	0.0003	7.25E-04	-4.0661	2.58E+00	-0.0096	1.94E-03
10	3	0.53	39.68	2.47	0.0673	7.15E-03	0.0257	1.40E-02	-0.0002	2.62E-04	0.5221	1.16E+00	-0.0078	1.01E-03
10	5	0.34	38.00	2.15	0.0675	6.59E-03	0.0201	1.34E-02	0.0002	4.02E-04	-2.9807	1.79E+00	-0.0093	1.26E-03
10	7	0.35	38.09	2.10	0.0681	4.13E-03	0.0229	8.14E-03	-0.0001	3.37E-04	-1.3016	1.17E+00	-0.0075	8.67E-04
10	12	0.40	36.76	1.54	0.0708	1.14E-02	0.0243	2.12E-02	-0.0003	4.61E-04	1.4674	1.89E+00	-0.0067	1.53E-03
10	13	0.42	38.71	1.83	0.0656	1.18E-02	0.0325	1.26E-02	-0.0006	7.49E-04	-1.2483	2.89E+00	-0.0073	1.55E-03

¹Change in normal acceleration from 1-g at a frequency of 1 radian per second²X-cg location measured in percent of mean aerodynamic chord³Average angle of attack for pass⁴Mean based on 52 samples⁵Confidence interval of the mean based on a 95 percent confidence level⁶Confidence level based on Desired Confidence Interval⁷Cramer-Rao bounds have been multiplied by a factor of 10

Table E.3. Turbulent Air Time History Match Cost Function Values

		Cost α ⁴	Cost q ⁵	Cost θ ⁶	Cost n_z ⁷
Mean ¹		16.88	4.49	5.82	21.90
Mean Confidence Interval ²		4.14	0.78	1.47	4.79
Standard Deviation		14.87	2.82	5.29	17.22
Parameter Weighting ³		23	3	14	60
Flight	Pass	Cost α	Cost q	Cost θ	Cost n_z
4	4	17.60	4.54	5.45	27.50
4	6	21.10	7.25	17.80	35.30
4	7	9.65	4.47	10.70	19.40
4	8	3.64	6.37	3.03	5.61
4	10	10.00	2.18	7.39	11.90
4	16	4.81	2.82	3.52	9.26
4	17	5.90	4.67	6.30	7.29
4	18	4.32	3.94	3.77	8.88
5	1	9.37	4.16	2.10	7.83
5	2	32.70	5.55	13.80	54.30
5	3	22.50	10.20	5.38	31.00
5	4	35.50	5.00	6.58	45.80
5	5	5.88	3.53	6.37	9.87
5	6	21.20	6.03	18.60	37.30
5	7	18.50	5.84	14.50	31.10
5	9	13.70	4.80	2.69	14.90
5	11	16.30	1.89	4.60	18.60
5	12	17.10	3.29	6.24	23.70
5	13	9.91	6.04	2.56	17.10
6	6	3.85	3.12	1.82	8.04
6	7	5.20	2.39	2.91	6.59
6	8	5.14	0.93	2.14	7.84
6	10	6.23	9.35	2.21	10.00
6	12	20.30	2.64	4.53	17.60
6	14	4.49	1.96	3.97	5.26
8	6	4.38	2.30	2.76	7.50
8	10	3.30	1.34	4.15	2.47
8	11	8.84	1.39	3.54	8.36
8	14	9.92	1.68	3.32	15.80
8	17	4.72	1.56	4.81	6.57
8	19	15.70	8.80	20.30	27.80
8	20	9.10	2.03	3.52	11.30
8	21	21.60	2.54	5.25	35.80
8	23	3.43	4.35	2.77	9.58
9	1	12.50	4.97	3.19	19.00
9	3	8.13	3.21	0.90	7.16
9	4	15.80	2.49	1.79	26.00
9	5	17.00	5.11	1.39	21.70
9	7	64.60	12.80	10.60	58.50
9	8	13.00	1.14	1.56	16.30
9	9	23.60	7.66	3.45	34.10
9	10	33.40	6.69	1.70	41.90
9	11	9.71	2.05	1.28	13.70
9	12	10.50	3.11	1.41	12.90
9	13	19.70	6.12	6.73	31.10
9	15	19.20	2.32	2.13	24.50
9	16	28.70	6.87	2.83	19.60
10	3	50.20	11.40	18.30	34.60
10	5	24.30	2.61	3.86	24.60
10	7	5.80	1.48	1.05	5.08
10	12	41.80	4.49	12.00	62.30
10	13	73.70	10.00	19.20	88.70

¹Mean based on 52 samples

²Confidence interval of the mean based on a 95 percent confidence level

³Mean parameter weighting from first ten calm air PCPID runs

⁴Cost function value associated with matching angle of attack measured time history

⁵Cost function value associated with matching pitch rate measured time history

⁶Cost function value associated with matching pitch angle measured time history

⁷Cost function value associated with matching normal acceleration measured time history

Bibliography

- Barnicki, Joe. GDSAVE. Computer file. Quartic Engineering Inc., Quartz Hill CA, 1996.
- Brigham, E. Oran. FFT, The Fast Fourier Transform and Its Applications. Englewood Cliffs NJ: Prentice Hall, 1988.
- Chalk, Charles R., Dante A. DiFranco, J. Victor Lebacqz, and T. Peter Neal. Revisions to MIL-F-8785B (ASG) Proposed by Cornell Aeronautical Laboratory Under Contract F33615-71-C-1254. AFFDL-TR-72-41. Wright-Patterson AFB OH: Air Force Flight Dynamics Laboratory, April 1973.
- Data Processing Branch, USAF Test Pilot School. Flight Test Analysis System (FTAS) User's Guide. Air Force Flight Test Center, Edwards AFB CA, January 1996.
- Department of Defense. Department of Defense Interface Standard, Flying Qualities of Piloted Aircraft. MIL-STD-1797A. Wright-Patterson AFB OH: Department of Defense (ASD/ENES), 30 January 1990, Notice 1, 28 June 1995.
- Flying Qualities Branch, USAF Test Pilot School. USAF TPS Flying Qualities Phase Text. Air Force Flight Test Center, Edwards AFB CA, December 1992.
- Grace, Andrew, Alan J. Laub, John N. Little, and Clay M. Thompson. Control System Toolbox User's Guide For Use with MATLAB®. Natick MA: The MathWorks Inc., July 1992.
- Hamel, Peter G. and Ravindra V. Jategaonkar. "Evolution of Flight Vehicle System Identification," Journal of Aircraft, Vol. 33, No. 1: 9-28 (January-February 1996).
- Hoh, Roger H., David G. Mitchell, Irving L. Ashkenas, Richard H. Klein, Robert K. Heffley, and John Hodgkinson. Proposed MIL Standard and Handbook - Flying Qualities of Air Vehicles, Volume II: Proposed MIL Handbook. AFWAL-TR-82-3081, Volume II. Wright-Patterson AFB OH: Air Force Flight Dynamics Laboratory, November 1982.
- Illiff, Kenneth W. Aircraft Parameter Estimation, AIAA Dryden Lecture in Research for 1987. NASA Technical Memorandum 88281. Edwards CA: National Aeronautics and Space Administration, 1987.
- Illiff, Kenneth W. and Richard E. Maine. Bibliography for Aircraft Parameter Estimation. NASA Technical Memorandum 86804. Washington: National Aeronautics and Space Administration, October 1986.
- Instrumentation Branch, USAF Test Pilot School. AT-37B Instrumentation Parameters. Air Force Flight Test Center, Edwards AFB CA, August 1996a.

- Instrumentation Branch, USAF Test Pilot School. USAF TPS Aircraft Instrumentation Handbook. Air Force Flight Test Center, Edwards AFB CA, August 1996b.
- Jones, Raymond L.. USAF Test Pilot School Curriculum Text, Volume II, Flying Qualities Phase, Chapter 15, Dynamic Parameter Analysis. Air Force Flight Test Center, Edwards AFB CA, December 1981.
- Kay, Steven M. Modern Spectral Estimation, Theory and Application. Englewood Cliffs NJ: Prentice Hall, 1988.
- Kelleher, James M. And David W. Milam. Flying Qualities and High Angle-of-Attack Evaluation of the F-16 A/B with the Increased Area Horizontal Tail. AFFTC-TR-82-12. Edwards AFB CA: Air Force Flight Test Center, August 1982.
- Klein, Vladislav. "Estimation of Aircraft Aerodynamic Parameters From Flight Data," Progress in Aerospace Sciences, 26: 1-77 (1989).
- Krauss, Thomas P., Loren Shure, and John N. Little. Signal Processing Toolbox User's Guide For Use with MATLAB®. Natick MA: The MathWorks Inc., June 1994.
- Ljung, Lennart. System Identification, Theory for the User. Englewood Cliffs NJ: Prentice Hall, 1987.
- Ljung, Lennart. System Identification Toolbox User's Guide For Use with MATLAB®. Natick MA: The MathWorks Inc., May 1995.
- Lockheed Fort Worth Company. Flight Manual, USAF/EPAF Series Aircraft F-16A/B Blocks 10 and 15. Technical Order 1F-16A-1. Hill AFB UT: Depart of Defense (OO-ALC/LFF), 14 August 1995, Change 3, 15 September 1997.
- Maine, Richard E. and Kenneth W. Iliff. User's Manual for MMLE3, a General FORTRAN Program for Maximum Likelihood Parameter Estimation. NASA Technical Paper 1563. Washington: National Aeronautics and Space Administration, November 1980.
- Maine, Richard E. and Kenneth W. Iliff. Identification of Dynamic Systems. AGARD-AG-300-Vol.2. Essex England: NATO Advisory Group for Aerospace Research and Development, January 1985.
- Maine, Richard E. and Kenneth W. Iliff. Identification of Dynamic Systems - Applications to Aircraft, Part 1: The Output Error Approach. AGARD-AG-300 Vol.3 Part 1. Essex England: NATO Advisory Group for Aerospace Research and Development, December 1986.
- MATLAB® for Windows. Version 4.2c.1. Computer software. The MathWorks Inc., Natick MA, 1994.

- Microsoft Word for Windows 95. Version 7.0a. Computer Software. Microsoft Corporation, Redmond WA, 1996.
- Microsoft Excel for Windows 95. Version 7.0a. Computer Software. Microsoft Corporation, Redmond WA, 1996.
- Mulder, J.A., J.K. Sridhar, and J.H. Breeman. Identification of Dynamic Systems - Applications to Aircraft, Part 2: Nonlinear Analysis and Manoeuvre Design. AGARD-AG-300 Vol. 3 Part 2. Essex England: NATO Advisory Group for Aerospace Research and Development, May 1994.
- Murray, James E. and Richard E. Maine. pEst Version 2.1 User's Manual. NASA Technical Memorandum 88280. Washington: National Aeronautics and Space Administration, September 1987.
- Nagy, Christopher J.. A New Method for Test and Analysis of Dynamic Stability and Control. AFFTC-TD-75-4. Edwards AFB CA: Air Force Flight Test Center, May 1976.
- Nagy, Christopher J.. Quartic Engineering Inc., Quartz Hill CA. Personal Correspondence. 21 August 1997a.
- Nagy, Christopher J.. Quartic Engineering Inc., Quartz Hill CA. Personal Correspondence. September 1997b.
- Newland, D.E.. An Introduction to Random Vibrations, Spectral & Wavelet Analysis. Essex England: Longman Group Limited, 1993.
- Operations Branch, USAF Test Pilot School. Modification Flight Manual, USAF Series Aircraft F-16A/B, S/N 80-0635, Air Force Flight Test Center, Edwards AFB CA, 30 Sept 1996.
- Pape, James K. And Michael P. Garland. F-16 A/B Flying Qualities Full Scale Development Test and Evaluation. AFFTC-TR-79-10. Edwards AFB CA: Air Force Flight Test Center, September 1979.
- Personal Computer Parameter Identification. Version 2.2. Computer software. Quartic Engineering Inc., Quartz Hill CA, 1996.
- Personal Computer Parameter Identification. Version 2.3. Computer software. Quartic Engineering Inc., Quartz Hill CA, 1997.
- Porat, Boaz. Digital Processing of Random Signals. Englewood Cliffs NJ: Prentice Hall, 1994.
- Quartic Engineering Incorporated. Personal Computer Parameter Identification (PCPID) User's Manual. Version 2.3. Lancaster CA: Quartic Engineering Incorporated, 1997.

- Ramirez, Robert W.. The FFT, Fundamentals and Concepts. Englewood Cliffs NJ: Prentice Hall, 1985.
- Roskam, Jan. Airplane Flight Dynamics and Automatic Flight Controls. Ottawa KA: Roskam Aviation and Engineering Corporation, 1979.
- Stambaugh, Robert M, and others. F-16B Parameter Estimation (PEST). AFFTC-TLR-92-12. Edwards AFB CA: Air Force Flight Test Center, May 1992.
- Signal Processing Toolbox For Use with MATLAB®. Version 3.0b. Computer software. The MathWorks Inc., Natick MA, 1994.
- Stevens, Brian L. And Frank L. Lewis. Aircraft Control and Simulation. New York: John Wiley and Sons, 1992.
- Welch, Peter D. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms," IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, No. 2. 70-73. New York: IEEE Press, June 1967.
- Wilson, Donald B. and Robert C. Ettinger. F-16 A/B High Angle of Attack Evaluation. AFFTC-TR-79-18. Edwards AFB CA: Air Force Flight Test Center, October 1979.

Vita

Captain Lawrence M. (Lars) Hoffman was born on 29 March 1966 in Valdosta, Georgia. He grew up as an Air Force dependent, graduating Valedictorian from Edwards Air Force Base's Desert High School in 1984. He subsequently attended the US Air Force Academy (USAFA) where he earned a Bachelor of Science degree in Engineering Sciences and a commission as a USAF second lieutenant upon graduation in June, 1988.

After graduation from USAFA, Lars completed Undergraduate Pilot Training at Williams AFB in June, 1989 and received a follow-on assignment flying the E-3 Airborne Warning and Control System aircraft out of Kadena AB, Okinawa, Japan.

In 1993, Capt. Hoffman completed Squadron Officer School in residence and earned an assignment as a U-2 high altitude reconnaissance pilot at Beale AFB, CA.

While earning a Master of Psychology degree in Human Factors Engineering from Idaho State University, Lars was selected for the joint Air Force Institute of Technology (AFIT) / US Air Force Test Pilot School (TPS) program. He enrolled at AFIT in May, 1995.

Capt. Hoffman completed all coursework toward a Master of Science degree in Aeronautical Engineering, specializing in aircraft stability and control and systems engineering while at AFIT, and continued research for the thesis requirement while attending TPS in 1997.

Lars graduated from TPS as an experimental test pilot in December, 1997 and is currently assigned to E-8 Joint Surveillance Target Attack Radar System Joint Test Force, Melbourne, Florida.

Permanent Address: 2067 Via Nuevo Leon
Green Valley, AZ 85614

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 1998		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE AN EVALUATION OF FREQUENCY DOMAIN ENSEMBLE AVERAGING TO IMPROVE AIRCRAFT STABILITY DERIVATIVE ESTIMATION			5. FUNDING NUMBERS	
6. AUTHOR(S) Lawrence M. Hoffman, Captain, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology 2750 P Street WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GAE/ENY/98M-02	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USAF Test Pilot School 220 S Wolfe Ave Edwards AFB CA 93524-5000 Comm: (805)277-3131 DSN: 527-3131			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Dr. Brad S. Liebst Comm: (937)255-6565 x4315 DSN: 785-6565 x4315 email: bliebst@afit.af.mil				
12a. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This research evaluated a process to improve aircraft stability derivative estimation results. The Have Derivatives process used overlap ensemble averaging in the frequency domain to minimize noise on the original time domain signals. The process estimated average complex frequency response functions that were then transformed back into the time domain as a set of discrete pulse responses with far less noise than the original signals. These 'clean' signals were used in a parameter estimation program to estimate better stability derivatives than were estimated with the original noisy signals. Both simulation and flight test data were used to study the effects of various noise levels on stability derivative estimation results and to evaluate the Have Derivatives process to improve those results. The simulations demonstrated dramatic improvement using the Have Derivatives process. The flight test results were not as conclusive. The ensemble averaging step of the Have Derivatives process was not effective enough at reducing noise on the flight test data due to non-uniform frequency content of the flight test input. The overall recommendation was to further evaluate the Have Derivatives process using a broadband flight test input, similar to the input that worked well in simulation.				
14. SUBJECT TERMS Parameter Estimation Spectral Estimation Aircraft Stability Derivatives Ensemble Averaging Signal Processing Output-Error Parameter Estimation Method Signal Noise System Identification Fast Fourier Transform Applications			15. NUMBER OF PAGES 181	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified
				20. LIMITATION OF ABSTRACT UL